VNIVERSITAT DE VALÈNCIA

Estimating Information In Earth System Data With Machine Learning

Autor: **Juan Emmanuel Johnson**

Directors: Valero Laparra i Gustau Camps-Valls

Doctorat en Enginyeria Electrònica

Març de 2021

Tesi Doctoral en Enginyeria Electrònica

Escola Tècnica Superior d'Enginyeria

Estimating Information in Earth System Data with Machine Learning

per

Juan Emmanuel Johnson

Directors:
Dr. Valero Laparra
Prof. Gustau Camps Valls

Doctorat en Enginyeria Electrònica
Universitat de València
Març-2021

Dr. Valero Laparra, Doctor en Informàtica i Matemàtiques Computacionals, Professor Ajudant Doctor al Departament d'Enginyeria Electrònica de l'Escola Tècnica Superior D'Enginyeria de la Universitat de València

Prof. Gustau Camps Valls, Doctor en Física, Professor (Catedràtic d'Universitat) al Departament d'Enginyeria Electrònica de l'Escola Tècnica Superior D'Enginyeria de la Universitat de València

Fan constar que:

Juan Emmanuel Johnson, BS in Mathematical Sciences, BS in Oceanography i MS in Applied and Mathematical Sciences, ha realitzat sota la seva direcció el treball titulat *Estimating Information in Earth System Data with Machine Learning*, que es presenta en aquesta memòria per optar al grau de Doctor per la Universitat de València.

I per tal què així conste a efectes oportuns, i donant el vistiplau per a la presentació d'aquest treball davant el Tribunal de tesi que corresponga, signem el present certificat a València el 24 de Març 2021.

Valero Laparra

Gustau Camps Valls

Tesi Doctoral:
Estimating Information in Earth System Data with Machine Learning

Autor:
Juan Emmanuel Johnson

Directors:
Dr. Valero Laparra
Prof. Gustau Camps Valls

El tribunal nombrat per jutjar la Tesi Doctoral citada anteriorment, compost per:

President: _____

Vocal: _____

Secretari: _____

Acorda otorgar-li la qualificació de _____

I per a què així conste a efectes oportuns, signem el present certificat.

A Burjassot el de de 2021

Note to the reader

According to the University of Valencia Doctorate Regulation[1] this PhD thesis is presented as a compendium of at least three publications in international journals containing the results of the conducted work. This thesis describes the published methods and the context within which they were developed. It also describes work that has recently been submitted to scientific journals. Furthermore, in accordance with the aforementioned regulation, and with the aim to foster the language of the University of Valencia in research and educational activity, the thesis also includes an extended abstract in Valencian.

[1] Reglament sobre depòsit, avaluació i defensa de la tesi doctoral aprovat pel Consell de Govern de 28 de Juny de 2016. ACGUV 172/2016.
Pla d'increment de la docència en valencià (ACGUV 129/2012) aprovat i modificat pel Consell de Govern de 22 de desembre de 2016. ACGUV 308/2016.

Acknowledgements

I would like to thank my advisors Gustau Camps-Valls and Valero Laparra for all of their support throughout this PhD endeavour. Without their expertise, guidance, and patience, none of this would have been possible. I would also like to give a massive thank you to my inner circle at the Image Signal Processing Laboratory: Anna, Jose, Nieves, Shari, Eatidal, Adrian, Diego, Roberto, Jordi C, Laura and Dan. They have been an absolutely fantastic group of peers for my entire 4 years both inside and outside of the lab. A special thanks to Daniel, Gonzalo and Emiliano for their support and for all of the adventures and laughs we shared together. I would also like to thank my many collaborators which have resulted in many, many fruitful discussions: Herve Claustre, José Enrique Adsuara, José J. Esteve-Taboada, Jesus Malo, Miguel D. Mahecha, Álvaro Moreno Martínez, Jordi Muñoz-Marí, Maria Piles, Adrián Pérez-Suay, Raul Santos-Rodriguez, and Kristoffer Wickstrøm. I would like to highlight and thank Katalin Blix, Ana B. Ruescas, and Raphaëlle Sauzède who provided me with the tether to ocean research throughout my time here which helped shape my future. A special thanks to the starry spotters team during my internship at FDL: Tansu Daylan, Lisseth Gavilan, Daniel K. Giles, Stela Ishitani Silva, Anna Jungbluth, Brett Morris, Andrés Muñoz-Jaramillo and Sairam Sundaresan. You all really taught me the value and mechanics of good teamwork which will stay for years to come. Lastly, I want to thank my family and loved ones for all of their love and support since my beginning.

Contents

Acronyms

AI Artificial Intelligence

ARD Automatic Relevance Determination

EO Earth Observation

CDF Cumulative Density Function

DGP Deep Gaussian Process

ELBO Evidence Lower Bound

GAN Generative Adversarial Network

GP Gaussian Process

GPLVM Gaussian Process Latent Variable Model

HMC Hamiltonian Monte Carlo

HSIC Hilbert-Schmidt Independence Criterion

ICA Independent Components Analysis

IT Information Theory

ITM Information Theory Measure/Metric

KDE Kernel Density Estimator

KECA Kernel Entropy Components Anaysis

kNN K-Nearest Neighbours

KRR Kernel Ridge Regression

KL-Divergence Kullback-Leibler Divergence

LHS Latin HyperCube Sampling

MC Monte Carlo

MI Mutual Information

ML Machine Learning

MLE Maximum Likelihood Estimation

MH Metropolis-Hasting

NF Normalizing Flow

NDE Neural Density Estimator

NUTS No U-Turn Sampler

PCA Probabilistic Components Analysis

PDF Probability Density Function

RBF Radial Basis Function

RBIG Rotation-Based Iterative Gaussianization

SA Sensitivity Analysis

SGP Sparse Gaussian Process

SOTA State-of-the-art

SGHMC Stochastic Gradient Hamiltonian Monte Carlo

SGLD Stochastic Gradient Langevin Dynamics

SVM Support Vector Machine

UQ Uncertainty Quantification

VAE Variational Autoencoder

wrt with respect to

XAI Explainable Artificial Intelligence

Nomenclature

Scalars, Vectors and Matrices

$N, D \in \mathbb{N}$	Natural number
$x, y \in \mathbb{R}$	scalar (real number)
$x_i \in \mathbb{R}$	the i-th sample from a collection of samples, X, where $(x_i)_{1 \leq i \leq N}$
$\mathbf{x} \in \mathbb{R}^D$	D-dimensional column vector, usually the input.
$\mathbf{y} \in \mathbb{R}^P$	P-dimensional column vector, usually the output.
$x^j \in \mathbb{R}$	the j-th feature from a vector, $\mathbf{x} \in \mathbb{R}^D$, where $(x^j)_{1 \leq j \leq D}$
$x_i \in \mathbb{R}$	the i-th sample from a vector, $\mathbf{x} \in \mathbb{R}^N$, where $(x_i)_{1 \leq j \leq D}$
$\mathbf{X} \in \mathbb{R}^{N \times D}$	a collection of N input vectors, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$, where $\mathbf{x} \in \mathbb{R}^D$
$\mathbf{Y} \in \mathbb{R}^{N \times P}$	a collection of N output vectors, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top$, where $\mathbf{y} \in \mathbb{R}^P$
$\mathbf{x}^j \in \mathbb{R}^N$	the j-th feature from a collection of vectors, $\mathbf{X}$, where $(\mathbf{x}^j)_{1 \leq j \leq D}$
$\mathbf{x}_i \in \mathbb{R}^D$	the i-th sample from a collection of vectors, $\mathbf{X}$, where $(\mathbf{x}_i)_{1 \leq i \leq N}$
$x_i^j \in \mathbb{R}$	the i-th sample and j-th feature from a collection of vectors, $\mathbf{X}$, where $(\mathbf{x}_i^j)_{1 \leq i \leq N, 1 \leq j \leq D}$

Common terms

θ	parameter or hyperparameter
$\boldsymbol{\theta}$	a collection of parameters or hyperparameters, $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_p]$
$f : \mathcal{X} \to \mathcal{Y}$	a latent function that operates on a scaler and maps a space $\mathcal{X}$ to a space $\mathcal{Y}$
$\mathbf{f} : \mathcal{X} \to \mathcal{Y}$	a latent function that operates on a vector and maps a space $\mathcal{X}$ to a space $\mathcal{Y}$
$k(\cdot)$	kernel or covariance function
$\lvert \cdot \rvert$	Determinant

Probability

$\mathcal{X}, \mathcal{Y}$	the space of data
P, Q	the probability space of data
$p_{\mathcal{X}}(\mathbf{x})$	the probability density function (PDF) on $\mathbf{x}$
$\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})]$	expectation of $f(\mathbf{x})$ over the distribution $\mathbf{x}$

Abstract

Machine learning has made great strides in today's science and engineering in general and Earth sciences in particular. However, Earth data poses particularly challenging problems for machine learning due to not only the volume of data, but also the spatial-temporal nonlinear correlations, noise and uncertainty sources, and heterogeneous sources of information. More data does not necessarily imply more information. Therefore, extracting knowledge and information content using data analysis and modeling is important and is especially prevalent in an era where data volume and heterogeneity is steadily increasing. This calls for advances in methods that can quantify information and characterize distributions accurately.

Quantifying information content within our system's data and models are still unresolved problems in statistics and machine learning. This thesis introduces new machine learning models to extract knowledge and information from Earth data. We propose kernel methods, Gaussian processes and multivariate Gaussianization to handle uncertainty and information quantification and we apply these methods to a wide range of Earth system science problems. These involve many types of learning problems including classification, regression, density estimation, synthesis, error propagation and information-theoretic measures estimation. We also demonstrate how these methods perform with different data sources including sensory data (radar, multispectral, hyperspectral, infrared sounders), data products (observations, reanalysis and model simulations) and data cubes (aggregates of various spatial-temporal data sources). The presented methodologies allow us to quantify and visualize what are the salient features driving kernel classifiers, regressors or dependence measures, how to better propagate errors and distortions of input data with Gaussian processes, and where and when more information can be found in arbitrary spatial-temporal data cubes. The presented techniques open a wide range of possible use cases and applications and we anticipate a wider adoption in the Earth sciences.

Resum

L'aprenentatge automàtic ha fet grans avenços en la ciència i enginyeria actuals en general i en les ciències de la Terra en particular. No obstant això, les dades de la Terra plantegen problemes particularment difícils per a l'aprenentatge automàtic a causa no només del volum de dades, sinó també per la presència de correlacions no lineals espacial i temporals alhora, d'una gran diversitat de fonts de soroll i d'incertesa, així com per la heterogeneïtat de fonts d'informació involucrades. Més dades no implica necessàriament més informació. Per tant, extreure coneixement i contingut informatiu mitjançant l'anàlisi i el modelatge de dades és crucial, especialment ara on el volum i l'heterogeneïtat de les dades augmenten constantment. Això requereix avenços en mètodes que puguen quantificar la informació i caracteritzar les distribucions i incerteses amb precisió.

Quantificar el contingut informatiu a les dades i els models del nostre sistema són problemes no resolts en estadística i l'aprenentatge automàtic. Aquesta tesi introdueix nous models d'aprenentatge automàtic per extreure coneixement i informació de les dades de la Terra. Proposem mètodes nucli ('kernel methods'), processos gaussians i gaussianització multivariant per tal de tractar la incertesa i la quantificació de la informació, i apliquem aquests mètodes a una àmplia gamma de problemes científics del sistema terrestre. Aquests comporten molts tipus de problemes d'aprenentatge, inclosa la classificació, regressió, estimació de densitat, síntesi, propagació d'errors i estimació de mesures teòriques de la informació. També demostrem com funcionen aquests mètodes amb diferents fonts de dades, incloses les dades provinents de distints sensors (radar, multiespectral, hiperespectral), nivells i productes (observacions, reanàlisi i simulacions de models) i cubs de dades (agregats de diverses fonts de dades espacial-temporals). Les metodologies presentades ens permeten quantificar i visualitzar quines són les característiques rellevants que governen distints mètodes nucli, com ara classificadors, mètodes de regressió o inclús les mesures d'independència estadística, com propagar millor els errors i les distorsions de les dades d'entrada amb processos gaussians, i on i quan es pot trobar més informació en espais arbitraris -cubs de dades espai-temporals. Les tècniques presentades obrin una àmplia gamma de possibles casos d'ús i d'aplicacions, amb les quals preveiem un ús més extens i robust d'algorismes estadístics en les ciències de la Terra i el clima.

Resumen

El aprendizaje automático ha hecho grandes avances en la ciencia e ingeniería actuales en general y en las ciencias de la Tierra en particular. Sin embargo, los datos de la Tierra plantean problemas particularmente difíciles para el aprendizaje automático debido no sólo al volumen de datos implicado, sino también por la presencia de correlaciones no lineales tanto espaciales como temporales, por una gran diversidad de fuentes de ruido y de incertidumbre, así como por la heterogeneidad de las fuentes de información involucradas. Más datos no implica necesariamente más información. Por lo tanto, extraer conocimiento y contenido informativo mediante el análisis y el modelado de datos resulta crucial, especialmente ahora donde el volumen y la heterogeneidad de los datos aumentan constantemente. Este hecho requiere avances en métodos que puedan cuantificar la información y caracterizar las distribuciones e incertidumbres con precisión.

Cuantificar el contenido informativo a los datos y los modelos de nuestro sistema son problemas no resueltos en estadística y el aprendizaje automático. Esta tesis introduce nuevos modelos de aprendizaje automático para extraer conocimiento e información a partir de datos de observación de la Tierra. Proponemos métodos núcleo ('kernel methods'), procesos gaussianos y gaussianización multivariada para tratar la incertidumbre y la cuantificación de la información, y aplicamos estos métodos a una amplia gama de problemas científicos del sistema terrestre. Estos conllevan muchos tipos de problemas de aprendizaje, incluida la clasificación, regresión, estimación de densidad, síntesis, propagación de errores y estimación de medidas teóricas de la información. También demostramos cómo funcionan estos métodos con diferentes fuentes de datos, provenientes de distintos sensores (radar, multiespectrales, hiperespectrales), productos de datos (observaciones, reanálisis y simulaciones de modelos) y cubos de datos (agregados de varias fuentes de datos espacial-temporales). Las metodologías presentadas nos permiten cuantificar y visualizar cuáles son las características relevantes que gobiernan distintos métodos núcleo, tales como clasificadores, métodos de regresión o incluso las medidas de independencia estadística, como propagar mejor los errores y las distorsiones de los datos de entrada con procesos gaussianos, así como dónde y cuándo se puede encontrar más información en cubos arbitrarios espacio-temporales. Las técnicas presentadas abren una amplia gama de posibles casos de uso y de aplicaciones, con las que prevemos un uso más extenso y robusto de algoritmos estadísticos en las ciencias de la Tierra y el clima.

List of Figures

List of Tables

1. Introduction

Contents

Extracting knowledge and information using data analysis and modeling is an important component. This is especially prevalent in an era where data volume and heterogeneity is steadily increasing. In this thesis we investigate machine learning methodologies to model and extract information in Earth system data. This chapter highlights the importance of uncertainty, information and knowledge extraction and introduces the necessary terminology and current approaches used to do so. It identifies some limitations within the literature and proposes concrete research objectives to overcome these limitations. This chapter concludes with an overview of the remainder of the thesis.

1.1. Motivation

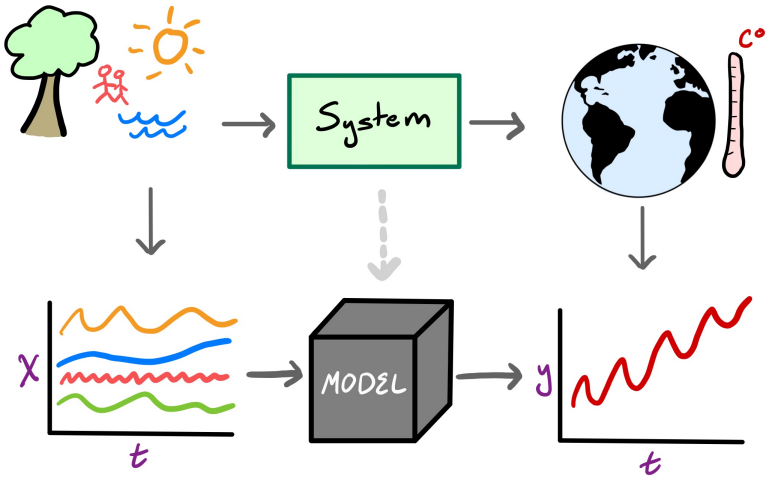


Figure 1.1.: A schematic of the nature of modeling and how all measurements and models are approximations of reality. There is an underlying true physical process that can describe the relationship between a set of input variables (e.g. biophysical, solar, oceanic and anthropogenic) and an output process (e.g. global temperature, sea level rise). Using the scientific method, we attempt to replicate that process using approximate models of reality. We feed these models using measurements and attempt to make predictions of the observations.

The Earth is a very complex system of interactions. Biological, chemical and physical sub-components all blend together to make up this dynamic and evolving Earth system. As scientists, our goal is to understand these interactions and to uncover the true underlying process behind the phenomena we observe. This is done in the form of modeling. A model is simply an abstraction of a true process which attempts to describe relationships, e.g. a single phenomenon or among multiple phenomena (Figure 1.1). Through abstraction, we can vary the complexity of our models by deciding which explainable variables are necessary to include in order to describe the phenomena in question. We can do analysis and characterize various aspects of the system that are observable by acquiring measurements of the system state. Then together, we can test and validate our assumptions via experimentation to find the best model given our measurements. We often attribute our progress within the last half century to two aspects: 1) better computational resources has allowed us to describe, process and experiment with many different model types and 2) more and better ways to gather measurements to facilitate all of the experimentation. Having these improved models and measurements have helped us make better predictions as well understand and discover new aspects of the Earth system.

One of the biggest challenges is dealing with the growing complexity and volume of our data necessary to validate and test our models. We have seen that traditional reductionist

learning approaches have stagnated over the years due to the sheer volume and heterogeneity of the data in conjunction with the ever increasing model complexity. We have recently seen an explosion of machine learning (ML) techniques to tackle many problems that were simply not feasible in the previous approaches. We often see many predictive or forecasting tasks that are present in our everyday lives such as autonomous vehicles, medical diagnosis and weather; all of which would not be possible without machine learning methods. In addition to better predictions, we have also seen new knowledge recovered from machine learning models. For example, drug discovery [Hudson, 2021, Senior et al., 2020], recovering equations of systems [Champion et al., 2019, Cranmer et al., 2020], and various geoscience applications [Reichstein et al., 2019, Bergen et al., 2019] all seek to provide us with completely new, and undiscovered results just from data. To gain insight into the models' decisions and extracting the learned information from the model, we have seen many approaches which involve post-hoc analysis of already trained models. These allow users to open the black-box of complex algorithms to get an intuition into the reasoning behind the model decisions. While some perspectives disenfranchise the use of complex black-box algorithms for simpler interpretable methods [Rudin, 2018], there is still a growing interest in uncovering the knowledge embedded within these black-box models [Miller, 2019, Lundberg et al., 2019, Samek et al., 2019]. This is the approach taken in this thesis as we look to exploit machine learning to help us uncover information embedded within the data thereby providing knowledge about the system.

What exactly is information? When we think of something as informative, it usually serves to answer specific questions *about* something. For example, in our day-to-day lives, we often consider something informative if it provides meaning to a question we may have asked. In other words, something is informative if we know more now than we did before we received this information. For example, we can have information *about* a certain quantity or a distribution of quantities. We could have information *about* the relationships between one set of phenomena and another set of phenomena. We could also have information *about* some constraints or assumptions about the state of a system. In the case of machine learning models, if one can quantify the information content within Earth observation data, then this would inform and enhance our fundamental understanding about how real systems operate. Quantifying the information content of the observational data and models, as well as their uncertainty levels is perhaps one of the most important challenges in standard statistics, modern machine learning and information theory. Observations and (ML) models form the windows we use to look at the world, and are the way we do inferences about the systems' behaviour. Real systems, like the Earth, are complex, networked and dynamic. Models are subject to all kind of noise sources, distortions and nonlinearities, and characterizing the model expressive power and trustworthiness are thus essential.

This thesis aims to provide motivated examples how we can use machine learning to extract knowledge and information in Earth observation (EO) data. EO is a particularly difficult field due to not only the volume of data, but also the spatial-temporal correlations and heterogeneous resources [Reichstein et al., 2019]. In the next section, we give a brief

overview of the concepts and methods used in this thesis. We outline some standard approaches and proposed solutions for how we can extract knowledge from our models which gives rise to concrete research objectives that will be tackled within this thesis. We conclude this chapter with a brief summary of the publications that resulted from this thesis and an overview of the structure of the remaining chapters.

1.2. Machine Learning Modeling Approaches

Let us assume we have a dataset, $\mathcal{D}$, as a pairwise set of input-output points $\{\mathbf{x}_i, y_i\}_{i=1}^N$. We can use the Bayesian formalism to define a model that takes into account uncertainty for all aspects of our models [Bishop, 2007, Murphy, 2012]. This provides us with a set of tools to describe $\mathcal{D}$ via its probability distribution $p(\mathcal{D})$. From a probabilistic modeling perspective, we are interested in a set of parameters $\boldsymbol{\theta}$ from a model $\mathcal{M}$ that we assume can describe the data $\mathcal{D}$, i.e. the posterior $p(\boldsymbol{\theta}, \mathcal{M}|\mathcal{D})$. We describe our parameter space with a prior distribution $p(\boldsymbol{\theta})$ about how the parameters should be distributed before we observe any data. Then we give a likelihood which describes the data generating process used to generate our data $\mathcal{D}$ given a set of parameters $\boldsymbol{\theta}$. Lastly, we marginalize over the evidence $p(\mathcal{D})$ by integrating out the parameters. Bayes rule gives us a formulation to link all of these quantities to find the posterior distribution

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}, \boldsymbol{\theta})}{\int p(\mathcal{D}, \boldsymbol{\theta}) d\boldsymbol{\theta}} = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}.$$

Broadly speaking, when dealing with data, we can approach a learning problem with supervised and unsupervised learning. Supervised learning takes some dataset, $\mathcal{D}$, as a pairwise set of input-output points and tries to learn some parameterized function, $\mathbf{f}_{\boldsymbol{\theta}}$, able to map the data from $\mathbf{x}$ to y. This effectively tries to model $p(y|\mathbf{x}, \boldsymbol{\theta})$ directly by solving a regression or classification problem. So now the problem has two sources of data uncertainty, the input and output uncertainty of $\mathbf{x}$ and y, respectively. Unsupervised learning considers the input dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ only, and tries to learn the joint distribution $p(\mathbf{x}|\boldsymbol{\theta})$ described by some function f parameterized by a set of $\boldsymbol{\theta}$. When modeling variables jointly, one can learn some hidden or underlying structure of the data itself. This is the holy grail of machine learning known as density estimation because, in principle, having the joint distribution gives us access to the marginal distributions $p(\mathbf{x}^1), p(\mathbf{x}^2)$ and the conditional distributions $p(\mathbf{x}^1|\mathbf{x}^2), p(\mathbf{x}^2|\mathbf{x}^1)$. Below we outline in more detail how does the machine learning community typically approach supervised/unsupervised learning and how does one typically account for the uncertainty.

1.2.1. Supervised Discriminative Modeling

The first case for supervised learning is a non-Bayesian approach that finds some parameterized function $\mathbf{f}_{\boldsymbol{\theta}}$ that maps from $\mathbf{x}$ to y (Figure 1.2.1). In this case, we take a complete

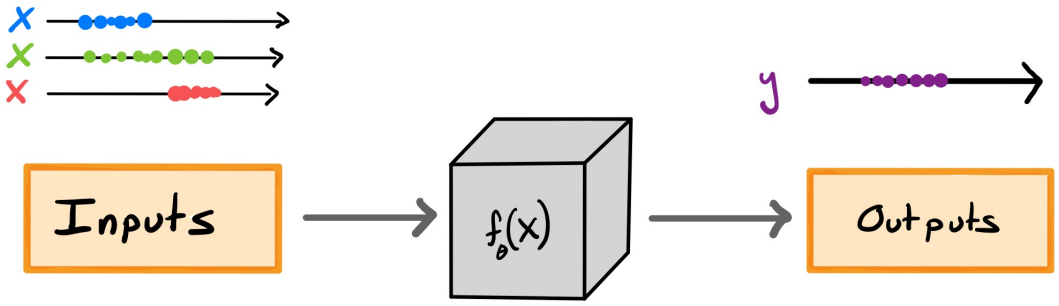


Figure 1.2.: A schematic of the discriminative machine learning methodology where we find the single best parameters $\boldsymbol{\theta}^*$ to describe the conditional distribution $p(y|\mathbf{x}, \boldsymbol{\theta}^*)$ which describes the relationship between $\mathbf{x}$ and y. Our input data comes from an underlying distribution $X \sim p_{\text{data}}(\mathbf{x})$ and we use the expected value of inputs, $\mathbf{x} = \mathbb{E}_{\mathbf{x}}[p_{\text{data}}(\mathbf{x})]$, as inputs to our model, $\mathbf{f}$. In addition, we assume that a single set of parameters, θ^*, can describe our approximate conditional distribution, i.e. $p(y|\mathbf{x}, \theta^*)$. This results in point-wise estimates of our output observations, $\mathbb{E}_{\theta^*}[p(y|\mathbf{x}, \theta^*)]$, of the true output variable, $p(y)$.

naive approach where we do not take into account the uncertainty in our model parameters or our data by finding the best single-set of parameters $\boldsymbol{\theta}^*$that best describe our data. First one needs to guess the parametric form of $\mathbf{f}$, e.g. a linear function, $f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$, a linear basis function, $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})$, or a non-linear function $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{g}(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}))$. Estimating $\boldsymbol{\theta}$ from $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ is called model fitting and it is easily the bulk of machine learning as we try to find a good and expressive representation of our data via some feature map $\mathbf{f}$ that minimizes the objective $\mathcal{L}(\boldsymbol{\theta})$ [LeCun et al., 2015b] . In the case of our supervised learning dataset $\mathcal{D}$ described by some likelihood function $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$, we could have the following loss function

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathcal{D}|\boldsymbol{\theta}) = \log p(y|\mathbf{x}, \boldsymbol{\theta}) + \lambda C(\boldsymbol{\theta}), \tag{1.1}$$

where λ is a regularization term and $\mathrm{C}(\boldsymbol{\theta})$ is a complexity term over the parameters $\boldsymbol{\theta}$ to mitigate overfitting. This particular formulation is known as Maximum Likelihood Estimation (MLE) as we assume that the expected value of the likelihood will give us the best set of parameters to describe our model $\mathbb{E}_{\boldsymbol{\theta}}[p(y|\mathbf{x}, \boldsymbol{\theta})]$. Although these are point estimates, we have seen incredible use cases in a wide range of applications [Graves et al., 2013, Krizhevsky et al., 2017, Young et al., 2018, Singh et al., 2020]. However, state-of-the-art (SOTA) discriminative models suffer heavily in the presence of uncertainty as they tend to be overconfident [Nguyen et al., 2015] and fail to generalize beyond their training data which give poorly calibrated mean estimates [Kawaguchi et al., 2017, Madry et al., 2018, Guo et al., 2019].

There are also many post-hoc methods in the literature which allow one to gain confidence by evaluating how the model has performed post-training. A classic method is to vary the inputs via targeted sampling strategies [Razavi and Gupta, 2015b] and propagate these through the function gradient, $\nabla_{\mathbf{x}^j} f$, to get a ranking of inputs most affecting the output variance. Then one can summarize the contribution of each input variable $\mathbf{x}^j$ as a contributing factor to the overall variance. An effective sampling strategy for the inputs, e.g. Random, Latin HyperCube [Helton and Davis, 2003] Quasi-Monte Carlo [Sobolá, 2001, Sudret, 2008], can give a good enough characterization to see how sensitive the model is to the input space. This sampling and gradient-based strategy is seen in other more modern methods which attempt including gradient-based feature attribution, surrogate modeling and perturbation methods [Jiménez-Luna et al., 2020]. The methods above are work-arounds to make models more robust to aleatoric and out-of-sample uncertainty. But they do not actually characterize the uncertainty in the data or the model parameters. This paints an incomplete picture as one should still incorporate uncertainty during the model training phase and within the model formulation to fully characterize the uncertainty in models and data.

1.2.2. Supervised Probabilistic Model

The parameter estimation method cannot take into account the uncertainty within our model parameters in its formulation due to its expectation on the parameter space. Instead of a single set of parameters $\boldsymbol{\theta}^*$, we get a distribution of that could describe the data $p(\boldsymbol{\theta}|\mathcal{D})$ (see Figure 1.2.2). From a supervised learning perspective, we are still assuming a function $f_{\boldsymbol{\theta}}$ that describes the relationship between $\mathbf{x}$ and y, i.e. $p(y|\mathbf{x}, \boldsymbol{\theta})$. So we can use Bayes rule

$$p(\boldsymbol{\theta}|\mathcal{D}) = p(\boldsymbol{\theta}|\mathbf{x}, y) = \frac{p(y|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(y|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} = \frac{p(y|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(y|\mathbf{x})} \tag{1.2}$$

to find the posterior $p(\theta|\mathbf{x}, y)$ given the prior distribution over the weights $p(\boldsymbol{\theta})$ and the likelihood $p(y|\mathbf{x}, \boldsymbol{\theta})$ describing the data generating process used to relate y and $\mathbf{x}$ given a set of parameters $\boldsymbol{\theta}$. The likelihood function depends on the task at hand, e.g. in a classification setting this would be a softmax likelihood while in a regression task, this could be a Gaussian likelihood or a T-Student likelihood. The density $p(\mathcal{D})$ is the *model evidence* or the *marginal likelihood* of the data which is a normalization term. For inference, we use the posterior which acts as a new prior for future data. We compute the likelihood given the parameters where each parameter is weighted by the posterior distribution. So given a new input $\mathbf{x}_*$, we can predict a new output y_*:

$$p(y_*|\mathbf{x}_*, \mathcal{D}) = \int p(y_*|\mathbf{x}_*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \tag{1.3}$$

This is a form of Bayesian model averaging which represents the average of infinitely many models $\mathcal{H}$ weighted by their posterior probabilities. The model provides an es-

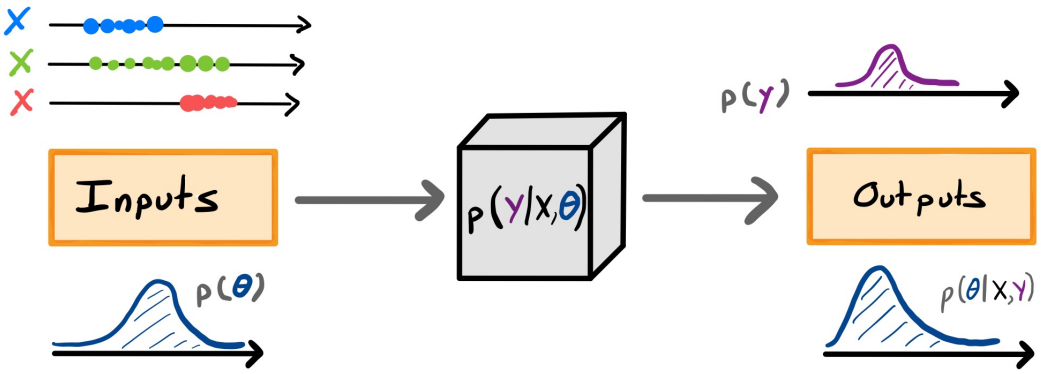


Figure 1.3.: A schematic of the probabilistic machine learning methodology where we find the distribution of the parameters θ that best suits the conditional distribution $p(y|\mathbf{x},\theta)$ which describes the relationship between $\mathbf{x}$ and y. Our input data comes from an underlying distribution $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ and we use the expected value of inputs, $\mathbf{x} = \mathbb{E}_{\mathbf{x}_i}[p_{\text{data}}(\mathbf{x})]$, as inputs to our model, $\mathbf{f}$. We assume that a distribution of parameters, $p(\theta)$, can describe our approximate conditional distribution, i.e. $p(y|\mathbf{x},\theta)$. Through Bayesian inference, one obtains a posterior distribution, $p(\theta|\mathbf{x},y)$, which describes the best set of parameters given our data as well as a predictive distribution for our output observations, $\mathbb{E}_\theta[p(y|\mathbf{x},\theta)]$, which approximates the true output distribution, $p(y)$.

timate of the epistemic uncertainty (the uncertainty in our model parameters; see Appendix A) over the possible parameters $\boldsymbol{\theta}$ that fit the data. This is in contrast to the discriminative approach which assumes a single set of parameters $\boldsymbol{\theta}^*$.

Inferring $p(\boldsymbol{\theta}|\mathcal{D})$ is the most difficult aspect of probabilistic modeling and is the dominant aspect of the literature [Bishop, 2007, Murphy, 2012, Ghahramani, 2015]. If the posterior is simple, then we can use exact methods to compute posterior distribution and in the case of likelihoods that are conjugate to the prior, then this can be done analytically. However, often times the true posterior $p(\boldsymbol{\theta}|\mathcal{D})$ is multimodal and complex and thus cannot be evaluated analytically so one has to choose alternative methods. There are deterministic methods which approximate a single or multiple nodes of $p(\boldsymbol{\theta}|\mathcal{D})$ with a convenient $q(\boldsymbol{\theta}|\mathcal{D})$ through optimization. These include methods like Laplace approximation [Mackay, 1995, Friston et al., 2007], variational inference [Beal, 2003], drop-out [Gal and Ghahramani, 2016], and expectation propagation [Minka, 2001]. Alternative methods are sample-based procedures which summarize the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ over a discrete set of samples. These methods are typically more asymptotically exact but can be slower than the approximate methods. Some algorithms include Metropolish-Hastings (MH) [Chib and Greenberg, 1995], Hamiltonian Monte Carlo (HMC) [Hoffman and Gelman, 2011, 2014], Stochastic-Gradient HMC (SGHMC) [Chen et al., 2014], and

Stochastic-Gradient Langevin Dynamics (SGLD) [Welling and Teh, 2011]. While originally Bayesian methods were critiqued for being much slower than parameter estimation methods like neural networks or ensemble methods [Lakshminarayanan et al., 2017], recent papers have showed that geometrically-inspired Bayesian methods [Izmailov et al., 2018, 2019, Maddox et al., 2019a, Yang et al., 2019] can provide better mean predictions and uncertainties while being more computationally efficient [Wilson and Izmailov, 2020].

1.2.3. Unsupervised Generative Modeling

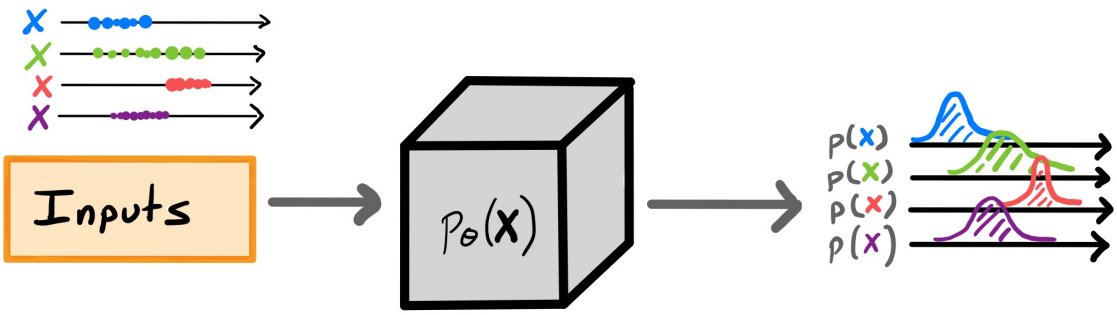


Figure 1.4.: A schematic showing the generative machine learning methodology where we find the best model $p_{\boldsymbol{\theta}}$ which can describe the $\mathbf{x}$. The input data comes from an underlying distribution $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ and we try to find an approximate distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ s.t. $p_{\text{data}}(\mathbf{x}) = p_{\theta}(\mathbf{x})$. We typically assume the process can be described by a function, $\mathbf{f}_{\boldsymbol{\theta}}$, parameterized by $\boldsymbol{\theta}$, which maps our data, $\mathbf{x}$ to a latent variable $\mathbf{z}$. Depending on the properties of the function, $\mathbf{f}$, one can generate samples and/or evaluate the density of new samples.

Unlike supervised learning, in generative modeling, we do not assume any mapping function $\mathbf{f}$ about the relationship between $\mathbf{x}$ and $\mathbf{y}$. Instead, we try to learn a generative model which tries to estimate the joint density of our data, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ (see Figure 1.2.3). Such a task can give us the underlying distributions of the data which could allow us to generate samples and/or evaluate densities. Formally, we assume that our data, $\mathbf{x}$, is multivariate and comes from some underlying distribution described by $p_{\text{data}}(\mathbf{x})$. In generative modeling, we want to learn some distribution $p_{\theta}(\mathbf{x})$ parameterized by $\boldsymbol{\theta}$ s.t. $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$. Generative models really are the holy grail of machine learning because many of the standard machine learning approaches are encapsulated within the generative modeling framework: compute conditional and marginal densities, compare datasets, dimensionality reduction, representation learning and data generation. However, unsupervised learning are the most difficult. Fortunately, a lot of progress has been made in the machine learning community in this aspect.

Just like with probabilistic predictive modeling, one needs to compute a model $\mathbf{f}_{\boldsymbol{\theta}}$ and then choose a learning principle to infer the parameters $\boldsymbol{\theta}$. One way is to introduce an unobserved random variable for every observed data point. This *latent variable* representation assumes a latent underlying distribution $\mathbf{z} \sim p_{\text{latent}}(\mathbf{z})$ that describes the distribution $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ under some transformation $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z})$:

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}), \qquad \mathbf{x} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{z}). \tag{1.4}$$

Note: in this case, we chose $\mathbf{z}$ to be a Gaussian distribution but in general any known distribution can be used, e.g. Uniform, Mixture of Gaussian's, etc. However, this formulation does not allow one to evaluate densities directly without some additional constraints on $\mathbf{f}_{\theta}$. If $\mathbf{f}_{\theta}$ is differentiable and bijective (diffeomorphic), then we can use the change of variables formula to evaluate the densities exactly. This is given by

$$p_{\text{data}}(\mathbf{x}) = p_{\text{latent}}\left(\mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{x})\right) \left|\boldsymbol{\nabla}_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{x})\right| \tag{1.5}$$

where $\boldsymbol{\nabla}$ is the Jacobian wrt $\mathbf{x}$, $|\cdot|$ is the determinant, and $\mathbf{f}_{\theta}^{-1}$ is the inverse function. Earlier methods assume some linear transformation, e.g. Probabilistic PCA [Tipping and Bishop, 1999] or Factor Analysis [Bishop, 2007]. To make these transformations more expressive, one could use a mixture of simple distributions, e.g. Gaussian Mixture model, which can arbitrarily approximate any distribution given enough components. One could also choose a non-linear transformation via neural networks [Mackay, 1995], Gaussian processes [Lawrence, 2005] or kernel methods [Schölkopf et al., 1998]. However, these methods all sacrifice invertibility for expressiveness as these functions are not bijective transformations. Generative Adversarial Networks [Goodfellow et al., 2014] are a popular family of models that use a neural network for $\mathbf{f}_{\boldsymbol{\theta}}$ and have shown the best results in generating samples that are qualitatively and quantitatively similar to $p_{\text{data}}(\mathbf{x})$. However, they are typically not bijective functions and thus one cannot evaluate densities. One can use hypothesis testing methods [Gretton et al., 2012, Dziugaite et al., 2015, Huszar, 2015] to compare distributions or use a minimax loss scheme that's predominantly used in GANs [Goodfellow et al., 2014].

Gaussianization [Chen and Gopinath, 2001, Laparra et al., 2011a, Meng et al., 2020a] and normalizing flows [Papamakarios et al., 2019b, Kobyzev et al., 2019] are the most common class of methods which use neural network architectures to find $\mathbf{f}_{\boldsymbol{\theta}}$ and only choose transformations that are differentiable and bijective. This class of methods are the most general of generative models because they allow one to generate samples and evaluate densities. In addition, since one can compute probabilities directly using the change-of-variables formula, one can simply use the maximum likelihood $\log p_{\theta}(\mathbf{x})$ to train these models. However, finding differentiable and bijective functions which are inexpensive to train can be difficult.

If one only wants to evaluate densities, then one can use separate neural networks to approximate the forward operation $\mathbf{f}_{\boldsymbol{\theta}}$ and inverse operation $\mathbf{f}_{\boldsymbol{\theta}}^{-1} = \mathbf{g}_{\boldsymbol{\theta}}$. This is the basis of the (variational) autoencoders [Wang et al., 2014, Kingma and Welling, 2014] which

feature a neural network for both the $\mathbf{f}_\theta$ and $\mathbf{g}_\theta$ (i.e. the decoder and encoder). This allows for arbitrary functions, $\mathbf{f}$, which are easier to evaluate. In standard autocoders, one has to minimize the reconstruction loss between the true data, $\mathbf{x}_{\text{true}}$ and the data generated from model $\mathbf{g}_x$, e.g. $||\mathbf{x}_{\text{true}} - \mathbf{x}_{\text{gen}}||$. So we force the latent representation to encode as much information as possible. Whereas in the variational autoencoders, one can factorize conditional distribution and marginalize over the latent variables. In this case, we are often faced with the standard integration problem within Bayesian framework (see the previous section).

1.3. Proposed Solutions and Limitations

We have outlined and decomposed three modeling perspectives used in machine learning. Depending upon the methodology, some approaches are more adapted for dealing with data and model uncertainty than others. However, each of the approaches are valuable and very well used within the machine learning community. We acknowledge the value of each of the learning paradigms and as such, consider each of the methods within this thesis. In the following subsections, we give some standard solutions for uncertainty quantification along with their limitations within each of the machine learning paradigms described in the previous section: supervised discriminative modeling, supervised probabilistic modeling and unsupervised gnerative modeling.

1.3.1. Sensitivity Analysis for Discriminative Modeling

In section 1.2.1, we gave an overview of how one can use machine learning modeling to estimate a single set of parameters which best describes the data. This is easily the most popular approach in machine learning. However, it inherently does not differentiate the aleatoric nor the epistemic uncertainty (the uncertainty within the data and the uncertainty within the model respectively; see Appendix A) without ad-hoc methods like ensembles. Admittedly, this class of methods are not the best suited for dealing with noisy data but they are still used abundantly within our community so it is important not to discount them. Fortunately, we can still characterize the inputs of our model on the output uncertainty through post-hoc analysis via sensitivity analysis.

Sensitivity analysis (SA) is one of the most popular analysis methods in the literature in Earth science models [Razavi and Gupta, 2015a]. It studies how the outputs of a model are related to the inputs. These inputs can be model parameters, initial and boundary conditions or model structures. The outputs can be any model response of interest which is often the predictions. Sensitivity analysis is a very effective framework which has been used to solve a wide range of questions. Some examples include 1) exploring how different processes and parameters affect the output of a system [Gupta and Razavi, 2018]; 2) to determine which influential factors are redundant and can be removed [Sobolá, 2001]; 3) allows one to assess the inputs that are dominant and also which parameters contribute the most to the output uncertainty [Guillaume et al., 2019, Iwanaga et al.,

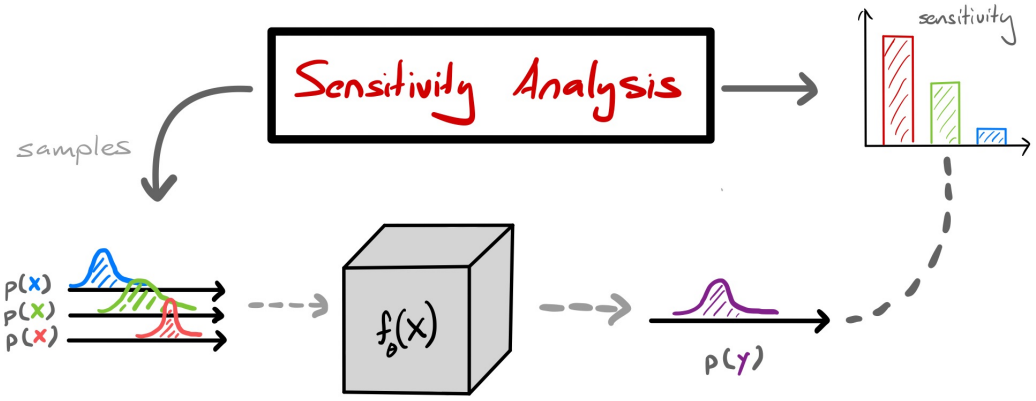


Figure 1.5.: A high-level overview of sensitivity analysis (SA) applied to a trained discriminative model $\mathbf{f}_\theta$. SA is a tool that generates input samples $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ that get propagated through a model $\mathbf{f}_\theta$ to produces some outputs y. Then it does some analysis of the variability of y with respect to the input $\mathbf{x}$ to characterize the relative influence $\mathbf{x}$ on the output variance of y.

2020]. Overall, SA helps practitioners make decisions about subsequent next steps to take regarding their model or data.

Sensitivity analysis can be divided into two subgroups: local and global. Local sensitivity analysis usually examines how the model output changes with small perturbations around a fixed point in the parameter space (Figure 1.5). Global methods take into account the entire parameter space. The simplest and most natural methods are gradient-based measures. This is a simple principle which postulates that high values given by the derivative of the model $\nabla_{x^j} \mathbf{f}_\theta$ can be attributed to a high sensitivity to the input sample x_i^k where $j \neq k$. To get a global measure, one can take the derivative with respect to a wider range of points and then summarize this by taking the mean, variance or some other statistic [Morris, 1991, Campolongo et al., 2007, Kucherenko and Iooss, 2014, Rakovec et al., 2014]. The gradient can be estimated with finite difference scheme or the adjoint method [Plessix, 2006] but this is very dependent on the step-size [Razavi and Gupta, 2015b]. With the advent of differentiable physical models and automatic differentiation, this limitation can be alleviated [Rackauckas et al., 2018].

However, sometimes one does not have access to the gradient explicitly as is the case with many physical models. Variance-based methods are a solution which seek to propagate distributions of input parameters and characterize the effect on the output variance. Monte Carlo sampling is the simplest and most exact method, yet very computationally

costly. However one can use quasi-Monte Carlo schemes [Lemieux, 2009] which sample from a pre-defined grid like Latin HyperCube Sampling (LHS) or Sobol indices [Sobolá, 2001] to achieve convergence with less iterations. Sampling methods can be expensive even with the quasi-MC methods. Another common option involves using surrogate models. These methods create an emulator which allow one to use a proxy ML model to replicate very expensive models that are cheaper and faster, e.g. polynomial chaos [Tennøe et al., 2018] or Gaussian processes [Camps-Valls et al., 2016]. Then one can characterize the response surface of the surrogate model as a proxy for the real physical model.

A lot of these methods above assume that the input features/parameters are independently distributed which is not always the case. Under this setting, one needs to construct a representative joint input distribution to sample from which is not always known a priori [Razavi and Gupta, 2015a]. This is difficult especially in the case of high-dimensional data and leads to a final class of methods which feature dependence metrics via dissimilarity measures. Instead of directly sampling and measuring the total variance of the propagated inputs, [Da Veiga, 2015] showed that one can model high-order relations dependencies between the input data [Székely et al., 2007, Gretton et al., 2012] which alleviates the need to construct ad-hoc distributions to sample from.

Sensitivity analysis in the context of uncertainty quantification (UQ) has shown a growing momentum in applied settings [Razavi et al., 2020] as it can be used to identify dominant factors within a system which can be useful for quantitive characterization and uncertainty reduction. With a good strategy, one could analyze the variance in the model predictions and potentially trace it back to the sources like the model components, model parameters and input data [Razavi et al., 2019]. A major issue with mathematical modeling is the increasing complexity of more complex structures which requires more components being added. In many real-world applications, this can lead to more "black-box" application of these models as some assumptions and parameters are "hard-coded" into the system [Mendoza et al., 2015]. In addition, this can misrepresent the full affect of the true influence [Colquhoun, 2014, Andrea et al., 2015, Nearing et al., 2016]. SA has the potential to help model developers calibrate the right level of complexity [Saltelli, 2019] to find the right balance between model completeness and propagation error [O'Neill and Rust, 1979, Turner and Gardner, 2015].

A critical challenge of SA is in the context of how can these methods be effectively used in the machine learning setting. Many SA methods used for physical models cannot be applied in the same way as ML models because they are constructed fundamentally differently. In a physical model, one can explicitly state a parameter that is responsible for representing a single process, e.g. temperature or salinity. In many complex non-parametric ML models, these methods do not assume a fixed parametric form as they instead try to estimate the function $\mathbf{f}_{\boldsymbol{\theta}}$ directly from the data. The parameters $\boldsymbol{\theta}$ for that function $\mathbf{f}$ often do not represent a process but instead an abstract representation of the data and uncertainty. So it might be difficult to single out a parameter that is responsible for a process. Instead, the value of SA is the ability to point to the most influential components and interactions between the input features of an ML model [Lundberg and

Lee, 2017a, Lundberg et al., 2019] which could lead to other applications such as model calibration for input variable selection or even model structure selection.

One of the biggest issues with sensitivity analysis is how one can connect traditional global sensitivity analysis to machine learning methods [Razavi et al., 2020]. Just like physical models, for ML methods, the sensitivity method will depend upon the properties of the model class. For example, ensemble methods and k-nearest neighbours do not have analytical gradients whereas kernel methods and neural networks do. No sensitivity criterion is universal and more attention should be paid to the inner mechanics of specific SA on targeted ML frameworks [Razavi and Gupta, 2015a]. Aside from importance score-based methods for feature ranking [Breiman, 2001, Lakshminarayanan et al., 2017, Weijs et al., 2010] and feature ranking methods [Blix et al., 2017, Rasmussen et al., 2011], little exploration has been done on the formulation and definition, nor the interpretation of sensitivity analysis of specific ML families. Overall, there should be a complete investigation of how we can calculate sensitivity measures within a model class to really bring forth clarity to the community about how we can use sensitivity analysis and why.

Research Question I: *Sensitivity analysis is one of the most popular analysis-methods used in forward physical models to characterize the impact of the inputs variables and model parameters on the output variance. However, in the context of machine learning,* ***understanding*** *and* ***interpreting*** *the results of this methodology is still a challenge due to the underlying differences in approaches. Can we formalize sensitivity analysis for the broad family of kernel methods?*

1.3.2. Probabilistic Modeling with Noisy Inputs

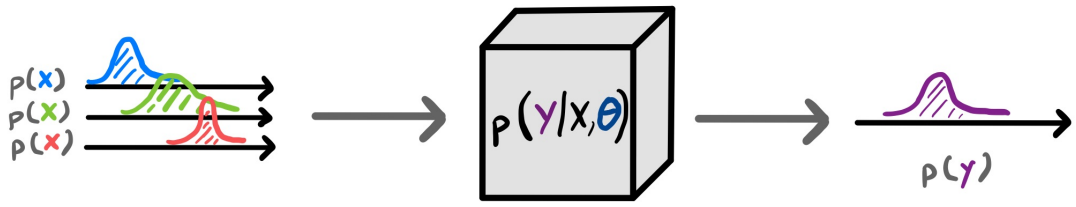


Figure 1.6.: A high-level overview of input error propagation through a probabilistic model.

In section 1.2.2, we gave an overview of how one can use probabilistic supervised machine learning methods to estimate the epistemic uncertainty using Bayesian methods. Using Bayesian methods gives us a complete probabilistic treatment of the system where we define priors of our system and extract the best distribution of parameters which

describe the model, i.e. the posteriors [MacKay, 2003, Gelman et al., 2020]. This is effectively inverse uncertainty quantification as we characterize our model parameters given the data. Then we can slowly iterate by critiquing and continuously improving our model assumptions by updating our priors given our new known posterior distributions (known as the Bayesian workflow [Gelman et al., 2020]) to obtain the best possible model given the data.

However, the standard Bayesian ML framework does not natively take into account the error in the inputs which could result in poor confidence intervals and even incorrect posteriors. In fact, Bayesian models accounting for input uncertainty are seldom found in the literature [Huard and Mailhot, 2006]. In traditional statistical models, accounting for uncertain inputs is known as error-in-variables [Kendall and Stuart, 1963]. There were a few competing ways in the literature of the relationship of how the inputs could be described: 1) we observe a noise corrupted version of the inputs, 2) we observe the actual inputs and assume the noise is independent. With either method, it's difficult to estimate the true inputs and model parameters. A deterministic approach was to use moment reconstruction [Freedman et al., 2004, 2008] which is an idea similar to regression calibration [Hardin et al., 2003]. More Bayesian approaches [Snoussi, 2003, Spiegelman et al., 2011] used a modified expectation maximization scheme and treated the inputs as hidden variables and [Dellaportas and Stephens, 1995] used Gibbs sampling to perform inference.

Gaussian processes are another popular class of probabilistic models which feature the golden standard confidence intervals even when compared to some of the recent Bayesian neural networks [Wilson and Izmailov, 2020]. Earlier studies looked at ways to augment the predictive mean and standard deviation by the Taylor's series expansion, Monte Carlo sampling and moment matching [Girard et al., 2002a]. These were mainly applied to dynamical models [Deisenroth and Mohamed, 2012] but also problems in hydrology to model streamflow [Sun et al., 2014]. We have also seen approaches for iterative technique to incorporate the noisy inputs into the training procedure [McHutchon and Rasmussen, 2011] which have been applied to improve modeling sea-level rise in historical records [Kopp et al., 2016]. However, despite the more recent progress of Gaussian processes (GPs) with input uncertainty, there is no literature (to our knowledge) that shows any application of these methods to many other Earth system related problems. Despite all of the advancements in the machine learning community and the importance it has in uncertainty quantification, it has received little to no attention in Earth systems data as there is a huge lack of literature showcasing how these modified GPs can be applied.

Research Question II: *Gaussian processes remain the golden standard for Bayesian machine learning which results in credible predictive mean and variances. In this thesis, can we demonstrate how to encode the uncertainty of our inputs into our GP models and quantify confidence gain in predictive uncertainty estimates.*

1.3.3. Generative Models for Information Theoretic Measures

In supervised learning scenarios, a model $\mathbf{f}$ is proposed to describe the relationship between the input $\mathbf{x}$ and output y, and the objective is to find a model f_θ given the input data. For both probabilistic models and for sensitivity analysis, we construct a set of assumptions or hypothesis $\mathcal{H}$ with the end goal of either explaining our predictions or capture our uncertainty. For example, one critical assumption in UQ is the choice of the multivariate properties of uncertain input variables $\mathbf{x}$ which are propagated through the model f. In SA, one assumes that they are independent and that each can be described through measurements, targeted experimental design or expert judgement. It has been shown that incorrect input spaces without considering the correlation structure can lead to poor results for variance-based sensitivity measures [Wang et al., 2018, Razavi et al., 2020]. In Bayesian modeling, we assume some distribution for our input variables $\mathbf{x}$ and the model for the generating process for y. But its also known that incorrect inductive biases and badly chosen priors (i.e. model mispecification [Dennis et al., 2019, DÁmour et al., 2020, Karaletsos and Bui, 2020, Wenzel et al., 2020]) lead to poor predictions. So how does one determine the best probability density function for the data? How does one assess which representation is more informative and which representation is more redundant? This still leaves us with a fundamental problem when we approach exploration and analysis from a modeling perspective: assumptions.

When we choose the model to represent the relationship between two datasets, we are inherently making assumptions about the form that the relationship takes. The data itself is its own representation of a state or phenomena in the world described in space and time. In section 1.2.3, we gave an overview of how one can use unsupervised probabilistic machine learning methods to directly model the data to gain insight into its underlying properties. Instead of modeling some relationship of $p(y|\mathbf{x})$, we directly model the joint density, $p(\mathbf{x}, \mathbf{y})$ where $\mathbf{y}$ can be multivariate $[y_1, y_2, \ldots, y_p]$. If we can capture the underlying joint density of our dataset $p(\mathbf{x}, \mathbf{y})$, this can potentially give us access to the conditionals, $p(\mathbf{x}|\mathbf{y}), p(\mathbf{y}|\mathbf{x})$ and also the marginals, $p(\mathbf{x}), p(\mathbf{y})$. In addition, having the joint density can improve our supervised models for example in Bayesian inference and data compression. So instead, we ask a different question: what can the data itself tell us? Can we establish some relevance or relationship between a variable and a phenomena? And how can we test and assess our assumptions?

> *"The basis for all knowledge is "information" that we compile about the world."*
> – [Kumar and Gupta, 2020]

The theory of information was presented in Shannon's seminal work [Shannon, 1948] which provides us with a framework that can quantify the uncertainty of our data. It is multivariate as it can provide quantities for systems of one or more variables, it is also model independent so we do not need to create hypothesis of what relationships *we think* the data has, and the units are also universal (bits) as it is not data dependent so we can compare across data types like discrete and continuous. The theory is deep and so

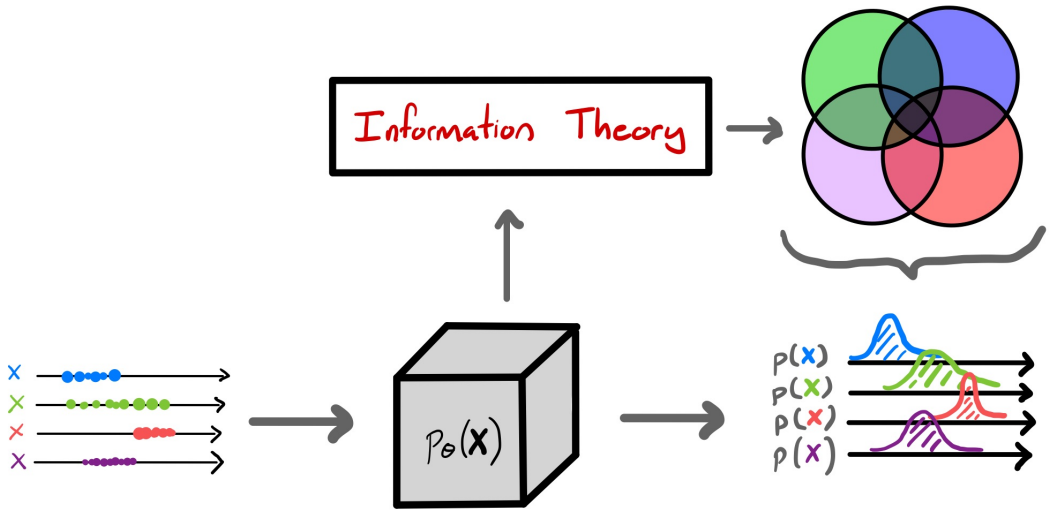


Figure 1.7.: A high-level overview of information theory and how it relies on density estimation to make comparisons across the individual variables with metrics like Shannon's interpretation of information, entropy, and total correlation.

we have a lot of metrics [Cover and Thomas, 2001] that can be derived from axioms of probability. There are many known ones; like 1) *entropy* - the average uncertainty of a distribution, 2) *mutual information* - the shared information between two or more variables, and 3) *relative entropy* - the asymmetric distance between two distributions; to other less known ones; like 4) *total correlation* - the redundancy of a distribution [Watanabe, 1960, Studený and Vejnarová, 1998] and *information flow* - conditional multivariate mutual information of three datasets [Petri, 1962]. Overall, information theory (IT) gives us a mathematical apparatus to assess whether the measurements or predictions reflect the true nature of the process [Kumar and Gupta, 2020].

Given the increase in complex data, information theory will be more prevalent in today's data analysis schemes in the cases where researchers do not have any assumptions about the underlying relationships between the data. However, subsequent analysis using these metrics can help one select more appropriate variables for modeling as it can also help us to *understand* the relationships between our datasets on a more fundamental level. This is especially powerful in scenarios of model building whereby decisions such as the prior distributions as well as the model architecture is decided based on the information content of the datasets. This framework enables one to quantify the information content within the data. In [Timme and Lapish, 2018], authors demonstrate that information theoretic measures can be an effective tool to analyze neuroscience data leading to better decisions about the modeling architecture.

"One variable provides information about another variable when knowledge of the first

reduces the uncertainty of the second."
–[Cover and Thomas, 2001]

Information theory metrics are scattered throughout the machine learning literature. For example, entropy is used in random forest schemes as a decision criteria, cross-entropy is often used as a loss function in classification tasks, mutual information for feature selection, and relative entropy (Kullback-Leibler Divergence) is often used in variational inference schemes. However, these are only metrics to assess a given criteria. Less emphasis has been placed on data analysis and UQ. In the Earth science community, information theory has not been widely used across communities but it does have a consistent use in some niche communities. Hydrology and ecology is by far the most common [Gong et al., 2013]. There is some earlier published work about the use of entropy in hydrology [Singh, 1997] and later perspectives about why information theory should be used [Weijs et al., 2010]. There is even a recent debate series where they argue that information theory is the new paradigm for science and argue that it overcomes some inconsistencies in the current Bayesian modeling approaches [Kumar and Gupta, 2020]. In climate science, it was first hinted that one should conduct exploratory data analysis and give summary *statistical descriptors*, ideally ones that were more expressive other than just correlation [Guttman, 1989]. Since then, there has been some studies to use information flow (a form of multivariate conditional mutual information) in climate-related applications [Bollt et al., 2018, Pot], more overviews and perspectives [Majda and Gershgorin, 2010, Ballantyne, 2016] as well as specifically identifying relevant climate variables [Knuth et al., 2013]. There has also been some efforts to improve how we visualize CMIP model comparisons using information theory measures [Correa and Lindstrom, 2013a]; a view that is also shared by [Larson, 2012].

However, information theoretic measures have not been widely accepted in the community due the limitations of *information theory estimators*. While the information theory measures are model independent, they are *data dependent* which means we do have to make some decisions about how to represent our data. All estimators depend on accurately estimating probability density functions, and this is difficult especially with moderate to high number of dimensions [Kumar and Gupta, 2020, Timme and Lapish, 2018]. As we increase the number of variables, the amount of data needed to capture their relations grows exponentially. This is especially prevalent in many Earth system applications with spatial, temporal and spectral features. For example, physical models modeling spatial distributions can take days or months to compute. Some physical models feature 100s or 1000s of parameters [Borgonovo and Smith, 2011, Lu et al., 2020] which makes SA or UQ very difficult. Even with the increasing computation resources [Prieur et al., 2019], this is still not feasible for everyone. Many times Earth and environmental systems modeling will use reduced-order models to reduce the dimensionality of the data [Sheikholeslami et al., 2019, McQuarrie et al., 2020, Kapteyn et al.]. However, one still needs to decide which dimensionality is correct and furthermore, generate samples that most-closely represent that distribution.

Both information theory and uncertainty quantification require accurate multivariate

densities, which is a long-standing problem in machine learning. Traditional parametric methods such as Gaussian mixture models and probabilistic PCA are not expressive enough to capture the data complexity within high-dimensional data. Standard non-parametric methods such as histogram, kernel density estimators and k-nearest neighbours approaches fail with datasets of this nature because they are susceptible to the curse of dimensionality. Generative modeling has increased in popularity and has had success with high-dimensional datasets like images. Ideally, one would need a method that is able to exactly model the density of the dataset as accurately as possible. GANs have shown to generate realistic samples but don't have access to the underlying density whereas VAEs are approximate models. Normalizing flows (NFs) have shown great promise in estimating very complex densities and high-dimensional datasets exactly [Papamakarios et al., 2019b, Kobyzev et al., 2019]. These methods use a parameterized function that can be trained jointly or iteratively [Laparra et al., 2011b, Inouye and Ravikumar, 2018] to find an invertible transformation from the original domain to a less complex domain. Then using the change of variables formula, one can estimate probability densities and sample effectively. [Chen and Gopinath, 2001] and [Laparra et al., 2011b] showcase how these class of models have a connection to IT. This makes this methods ideal for not only estimating densities but also for estimating information theory metrics for high-dimensional datasets. However, this link has yet to be fully explored especially within the context of Earth system data.

Research Question III: *Information theory metrics provide us with a framework for summarizing the expected uncertainty in our data. While there are many advantages of information theory for data characterization, it is incredibly difficult in high-dimensional, multivariate datasets. Can we address this inherent limitation and demonstrate its viability in real world applications?*

1.4. Research Objectives

The scientific machine learning community has made great strides to improve physics-aware modeling but lesser attention has been paid to the uncertainty quantification of observational data and its adverse impacts on the quality of our models. In particular, we stated that uncertainty in our input data is a key consideration for all facets of machine learning however this is often not considered in standard practice. A lot of data is not necessarily informative data so it is important to ensure that our data is correctly characterized in order to achieve the best possible and most correct model representation.

"The goal of this thesis is to introduce new ways of quantifying uncertainty and information content within Earth system data and models."

Table 1.1.: A summary of all components in the research objectives within this thesis: 1) for **supervised discriminative** modeling - we look at derivative-based sensitivity analysis to analyze kernel methods; 2) for **supervised probabilistic** modeling - we improve Gaussian processes confidence intervals via error propagation of noisy inputs; 3) for **unsupervised generative** models - we use Gaussianization as a generative model for density estimation and Information theory estimation for high-dimensional, multivariate data.

Algorithm Class	Assumptions	Analysis	EO Application
Kernel Methods	$p(y\|\mathbf{x}, \boldsymbol{\theta}^*)$	Sensitivity Analysis, $\nabla_{x_j} f(\mathbf{x})$	Feature Relevance
Gaussian Processes	$p(y\|\mathbf{x}, \boldsymbol{\theta})$	Uncertainty Propagation, $p(y\|\mathbf{x}, \boldsymbol{\theta})p(\mathbf{x})$	Regression, Kriging, Sampling
Gaussianization	$p(\mathbf{x}\|\boldsymbol{\theta})$	Information Theory, (e.g. Entropy, mutual information)	Feature Relevance, Density Est., Info. Content

Why is the topic important?

Acknowledging and adjusting for noisy data is especially relevant now when we have increasing amounts of data from heterogeneous and secondary sources. In addition, we now have copious volumes of spatial-temporal data which may or may not be informative for our applications. However, machine learning methods applied to Earth science data are often used without taking input data uncertainty or characteristics into account. By acknowledging and accounting for this uncertainty and inherent structure within our input data through direct quantification or explainable measures, users will have more confidence in our models which will lead to better and more informed decisions. This will not only help our predictive performance, but it will also help our understanding of the underlying data characteristics.

How do we plan address this topic?

For three machine learning disciplines, we outlined the literature for a subset of the standard methodologies used for modeling and highlighted potential pitfalls and limitations surrounding these approaches to account for the uncertainty within the data. In parametric learning, sensitivity analysis is the standard approach but the discrepancies between how it is applied to physical models and non-parametric machine learning is still an unexplored area. We will show that SA is a viable and interpretable framework that can be applied to an entire class of machine learning models, specifically under the framework of kernel methods. In supervised probabilistic modeling, we highlight that Bayesian models are excellent at accounting for epistemic uncertainty but often fall short when considering the aleatoric uncertainty. We will demonstrate that accounting for uncertain inputs via

error propagation can improve the predictive uncertainty in a class of ML models, specifically under the framework of Gaussian processes. In unsupervised learning, we highlight information theory provides excellent measures of overall data uncertainty but it is not widely adopted due to the problem of density estimation in high dimensional spaces. In this thesis, we will demonstrate that obtaining IT metrics is possible even for high dimensional data via a class of normalizing flow models called Gaussianization. In all cases, we will focus on spatial, temporal and/or spectral EO data. In the next subsections, we give the concrete steps taken to address the highlighted topics (see Table 1.1).

1.4.1. Objective 1: Sensitivity Analysis for Kernel Methods

Summary. We fully explore derivative-based sensitivity analysis in Kernel methods to open the black-box on an entire class of algorithms from regression, classification, density estimation and dependence estimation.

Idea. To provide intuition on how one might use sensitivity analysis for kernel methods, we go through the kernel formulation and show how (like linear methods) kernel methods have an analytical derivative which means one has access to derivative-based analysis methods even though they are a non-parametric class of algorithms. We verify this by giving a complete end-to-end exploration of derivative-based sensitivity analysis and related methods focusing on interpretability. We highlight how this works with not only standard regression methods, but also can be extended to classification, density estimation and measuring dependence methods.

Application. We focus on Earth system data exclusively. The data is spatial-temporal and we give example applications of how we can use SA to gain insight into the fitted kernel method. We demonstrate this using regression, classification, density estimation and dependence estimation. This will further motivate the field to use this framework as it will stem directly from the explanation of the theory mentioned above.

Relevance. By choosing kernel methods for derivative-based sensitivity analysis we provide the following contributions: 1) we counter the idea that ML models are black-boxes especially kernel methods, 2) we validate and verify the application of derivative-based sensitivity analysis for a class of ML-models, and 3) we generalize this methodology outside of the standard regression/classification schemes to kernel-based density estimation and independence estimation. In all cases, this pushes the field forward for better uncertainty quantification and explainability for kernel methods in particular.

1.4.2. Objective 2: Uncertain Inputs for Gaussian Processes

Summary. We improve the confidence intervals of Gaussian processes by accounting for uncertain inputs.

Idea. We take inspiration from dynamical GP models and show how one can incorporate

input uncertainty into the standard GP formulation by simply modifying the predictive mean and predictive variance. We emphasize that even a simple modification via the gradient of the GP predictive mean can lead to more qualitatively and quantitatively better confidence intervals.

Application. We use noisy, high-dimensional satellite reflectance data from a hyperspectral sensor used to predict global temperature as our case study. This dataset has been used in previous applications, however none of the practitioners incorporated the error in the spectral bands directly within their ML models. The quality of the confidence intervals for the global map are assessed.

Relevance. Gaussian processes are a very popular class of Bayesian methods and remain the golden standard for sensible confidence intervals. However, we postulate that the aleatoric uncertainty that isn't accounted for within our inputs leads to sub-optimal confidence intervals. By demonstrating that even the simplest of methods can improve the confidence intervals, this will motivate the community to take further steps to improve the widespread use of GPs for EO applications.

1.4.3. Objective 3: Gaussianization for Density Estimation and Information Theory Metrics

Summary. Showcase the use of generative models for high-dimensional density estimation and information theory metrics computation in Earth system applications.

Idea. We acknowledge that information theoretic measures have not been adopted in the standard literature due to issues with PDF estimation for high-dimensional datasets. We address this limitation by using Gaussianization, a generative model within the family of normalizing flows. We also demonstrate that the real issue of UQ (in the pure sense of the word) can be addressed via information theoretic metrics as a superior metric than standard correlation measures. We also show why Gaussianization is an ideal choice due to the close connections to IT within the formulation.

Application. We give many toy examples like radar and hyperspectral images with complex distributions to demonstrate that Gaussianization can correctly generate samples that are marginally and jointly representative of the original dataset. We showcase Gaussianization as an ideal tool for exploring large-scale spatial-temporal Earth system global data products where we give examples of how information theory metrics, like entropy and mutual information, can be used to assess the optimal configuration for further processing.

Relevance. Information theory metrics are powerful but underutilized due to the curse of dimensionality which also explains the ineffectiveness of standard density estimation models. In this portion of the thesis, 1) We introduce a competitive model for density estimation into the Earth science community, 2) we showcase how information theory metrics are now a viable option for analyzing high-dimensional, multivariate datasets,

and 3) by directly characterizing the joint distribution via machine learning models.

1.5. Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 is relevant to research objective 1 outlined in 1.4.1. It presents the basics of sensitivity analysis to give context for the included paper. We also provide a case for kernel methods to motivate why we chose that particular class of methods for machine learning methods. This is followed by a short summary of the article, contributions, impact and reproducibility is provided. This chapter is concluded with parting thoughts for future work.

Chapter 3 is relevant to research objective 2 outlined in 1.4.2. It presents a short summary of probabilistic modeling followed by an overview of Gaussian process regression. I provide a history of previous work for incorporating uncertain inputs into GPs with toy examples for each of the core methods listed. This is followed by a short summary of the article, contributions, impact and reproducibility is provided. This chapter is concluded with further literature building upon the published work with a list of concrete steps for future work.

Chapter 4 is relevant to research objective 3 outlined in 1.4.3. It briefly outlines the SOTA generative modeling framework motivating the choice of methods. It further motivates the Gaussianization framework over the previously described methods via its connection to information theory. This is followed by a short summary of the paper, contributions, impact and reproducibility. This chapter is concluded with further literature building upon the published work with a list of concrete steps for future work.

Chapter 5 provides an overall discussion and conclusion of the thesis where we recap the motivation, research, objectives, summaries of the papers a synergistic overview of the future work.

Appendix. This contains material useful for each of the chapters to further the readers understanding but is not directly relevant to this thesis nor the methodology section.

2. Sensitivity Analysis for Kernel Methods

Contents

Kernel methods span across many disciplines including regression, classification, density estimation and dependence measures. However, it is often said that they are black-box models. To combat this, we use derivative-based sensitivity analysis and its principles to showcase how these models are indeed interpretable and can be used not only for modeling, but also to improve our understanding of the underlying data characteristics and relevant features. We demonstrate the use of powerful analysis techniques using complex Earth system data with targeted examples. The work presented within this chapter serves as a building block for future analysis to promote sensitivity analysis as a viable solution for explainable machine learning.

2.1. Sensitivity Analysis

The sensitivity of the output y given an input $\mathbf{x}$ that function $\mathbf{f}$ has a clear definition: it characterizes changes in the output based on propagating local perturbations of the input space through the partial derivatives of the function [Razavi et al., 2020]. Let $\mathbf{f}$ be a function s.t. $\mathbf{f} : \mathbf{x} \in \mathbb{R}^D \rightarrow y \in \mathbb{R}$. It is rather intuitive when we consider gradients in general. We can define a sensitivity index s_j

$$s^j(\mathbf{x}_i) = \nabla_{\mathbf{x}_i^j} \mathbf{f}(\mathbf{x}_i) \tag{2.1}$$

where j is the j-th feature of the multivariate input $\mathbf{x}_i$. If the gradient of the function, $\mathbf{f}$ wrt to the input feature, $\mathbf{x}^j$ is equal to zero, i.e. $\nabla_{\mathbf{x}^j} \mathbf{f}(\mathbf{x}_i) = 0$, then the point $\mathbf{x}_i$ is located on a plateau of y and the uncertainty associated with that variable is 0. This is because any perturbations around $\mathbf{x}_i$ will not have any changes in the output y_i. However, if $\nabla_{\mathbf{x}^j} \mathbf{f}(\mathbf{x}_i) \gg 0$, then the point $\mathbf{x}_i$ is located on a steep plane of y where small variations of $\mathbf{x}$ can lead to large variations in y_i. This is often defined *local sensitivity* because they rely on the linearization of the model around a defined set of points of interest and a sample space of possible values.

To do a global summary of these derivative-based global sensitivity analysis (GSA) methods, one can summarize the contributions via the first and second moments, e.g. the mean and the variance, given the matrix of inputs, $\mathbf{X} \in \mathbb{R}^{N \times D}$, such that $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$. So an example is

$$\bar{s}^j(\mathbf{X}) = \int_{\mathcal{X}^j} \left(\nabla_{\mathbf{x}^j} \mathbf{f}(\mathbf{X})\right)^2 p(\mathbf{x}^j) d\mathbf{x}^j \approx \mathbb{E}_{\mathbf{x}_i} \left[\left(\nabla_{\mathbf{x}^j} \mathbf{f}(\mathbf{X})\right)^2 \right]. \tag{2.2}$$

One could use other summary statistics such as the absolute value or the root-mean-squared. This is the basis of elementary methods like [Morris, 1991] and its subsequent extensions [Campolongo et al., 2007, Rakovec et al., 2014, Kucherenko and Song, 2016]. One issue which has been explored by [Razavi and Gupta, 2015b] is how to use these methods when the analytical gradient is not available. In this case, one needs to use approximate methods like finite differences which can be very dependent on the step-size. It has also been shown that there is a direct connection between derivative-based GSA and variance-based GSA like Sobol's indices [Sobolá, 2001, Razavi et al., 2019]. See [Saltelli et al., 2008, Pianosi et al., 2016, Borgonovo and Plischke, 2016, Ancona et al., 2017, Tennøe et al., 2018, Douglas-Smith et al., 2020, Razavi et al., 2020] for surveys on other sensitivity analysis methods.

2.2. Case for Kernel Methods

Data representation is an important aspect of machine learning. When we think of how machine learning models learn, they learn a representation of the data as a way to reveal

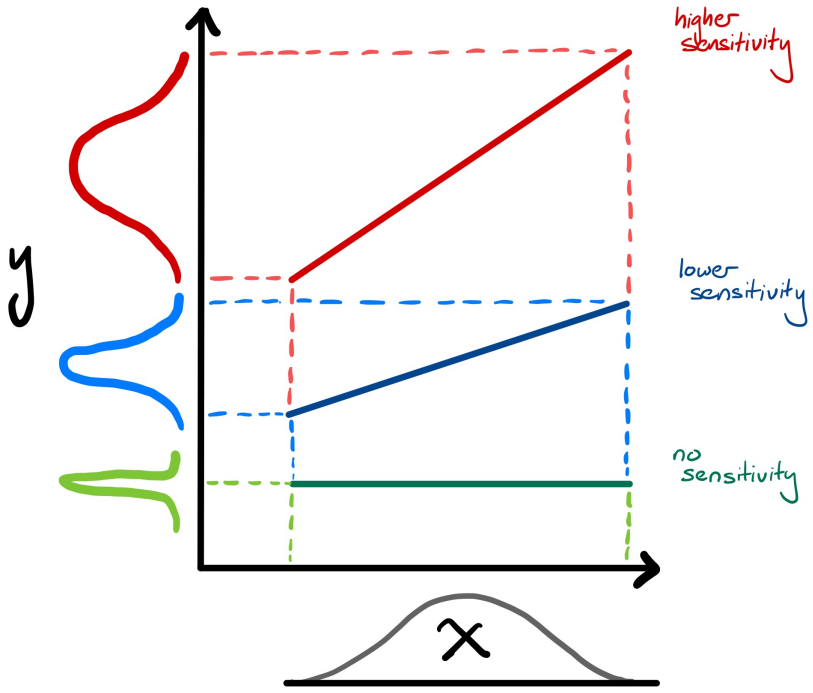


Figure 2.1.: This shows an example of local sensitivity where the gradient gives us insight into how much the outputs will vary given perturbations within the inputs. In this example, we see that perturbations in the input data, $\mathbf{x}$, lead to not much change in output data, y where the gradient is zero (green). However, there are large changes where the gradient is greater than zero (red). Figure adopted from [Loucks and van Beek, 2017]

interesting patterns and aid in predictions. In the case of many learning problems regression, classification of dimensionality reduction, the simplest representation is a linear function:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}^\top \mathbf{x} \tag{2.3}$$

While a linear transformation is the simplest and most interpretable model, many real world applications have datasets which exhibit non-linear relationships between the features and therefore one needs models that exploit non-linear feature representations. So many machine learning methods spend a lot of time finding some mapping function $\phi(\cdot)$ such that the the linear operation is simple

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle \tag{2.4}$$

If we have a good representations of our data via some feature map [LeCun et al., 2015a] then this function would be able to recognize patterns. So essentially, if a good feature

representation is found, one can translate non-linear datasets into linear representations, decompose factors into independent components, and separate many classes into linearly separable classes. Thus, the machine learning field primarily investigates different ways to represent the data such that one can represent the data linearly.

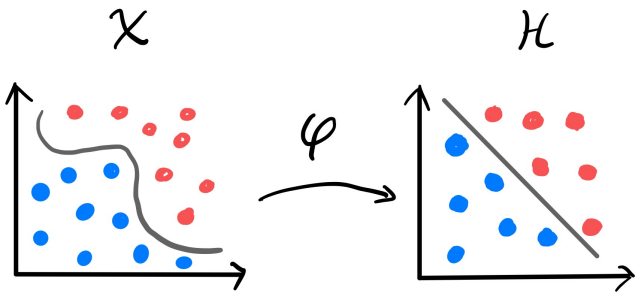


Figure 2.2.: A simple illustration showing the basic idea of kernel methods: we choose a suitable map $\phi(\mathbf{x})$ for the data embedded in $\mathcal{X}$ s.t. they are linearly separable in the space $\mathcal{H}$.

One of the most diverse solutions available which gives us diverse feature representations in the literature are kernel methods [Schölkopf and Smola, 2002a, Raamana, 2020]. This class of methods allows one to transform most linear methods into non-linear methods by the use of a kernel function, while still resorting to linear algebra operations. This works by mapping the data from the input space $\mathbf{X} \in \mathbb{R}^{N \times D}$ to a higher dimensional space $\mathcal{H}$ using a kernel function k to construct a similarity matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$. With a good mapping function $\boldsymbol{\phi}$ and its corresponding kernel function k, one can find an embedding in which the data is now linearly separable. Similar to the above equation, we can write $f(\mathbf{x})$ in terms of k as:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{2.5}$$

Consequently, we have seen many traditional linear methods utilize this kernel trick to deal with complex data. There is no doubt that this is a powerful set of methods. See [Schölkopf and Smola, 2002b, Shawe-Taylor and Cristianini, 2004, Rojo-Alvarez et al., 2018] for complete overviews. Kernel methods are universal learners capable of learning many arbitrary representations given enough data and the correct kernel function. This is a very attractive property as it provides a flexible model class to naively apply and get good results. Another advantage is that kernel methods are not limited to vectorial data. Many times we do not always have point estimates of our datasets. We often have other ways to present our data, e.g. graphs, probabilistic distributions, manifolds and time series. The kernel methods literature is vast so there is essentially a kernel function that can be specified even for heterogeneous data types as inputs. From there, a kernel

matrix is simply a similarity of the N inputs such that we have an $N \times N$ representation of the similarity in the data. Lastly, one needs to find an objective function (if there is one as this depends upon the task) so that we can maximize/minimize the parameters of the kernel function that coincides with the objective function. One final advantage is that of combining kernels to get more expressive and flexible solutions: for instance, one could also define a kernel per data type and per task, and then find a way to aggregate the results. This is known as multiple kernel learning (MKL) [Rakotomamonjy et al., 2008, Camps-Valls et al., 2006, Gönen and Alpaydin, 2011]. In machine learning, we often have different tasks which expect different data representations and so having MKL as an option can boost the efficacy of kernel methods for multi-modal tasks [Sonnenburg et al., 2006, Xu et al., 2013, Baltrusaitis et al., 2019].

Small Datasets

Not all machine learning methods can cope with big data problems. Often times there are many applications where we have small datasets of eventually high dimensionality. Kernel methods excel in this area by exploiting the duality of the sample representation versus the feature representation [Schölkopf and Smola, 2002a]. For example, in medical datasets [Shaikhina and Khovanova, 2017], we often have more features than samples. Ensemble methods are a poor choice because there are not enough samples to do boosting and NNs are not a good choice because they require a lot of data to learn good feature representations. On the other hand, Kernel methods have shown excellent performance on smaller datasets [Cheng and Kingsbury, 2011, Lu et al., 2014, Dai et al., 2014, Huang et al., 2014, May et al., 2019]. There are also Bayesian extensions to all of the methods which can be beneficial in small data problems due to probabilistic assumptions and data descriptors. In the case of kernel methods, this allows one to specify functions parameterized by kernel functions which give an infinite basis as is the case for regression, classification, dimensionality reduction [Bishop, 2007, Murphy, 2012]. Alternatively, one can use probabilistic methods to construct kernel matrices which can provide a better fit for the data [Izquierdo-Verdiguier et al., 2015, Muandet et al., 2017, Løkse et al., 2017].

Scale

Kernel methods are non-parametric similarity measures which construct a matrix with the full dataset. So for massive amounts of data (10000+), they are computational expensive. Traditionally, one used the literature regarding matrix approximation to address this problem. Kernel functions produce matrices that can be approximated with data dependent methods such as count-sketching [Wang et al., 2015, Zhang and Liao, 2019], random projections [Sánchez et al., 2018, Paul et al., 2014] or the Nyström approximation [Williams and Seeger, 2000]. An alternative approach is to approximate a kernel function with a data independent randomized function [Rahimi and Recht, 2007, Le et al., 2013, Yu et al., 2016]. See [Yang et al., 2012] for a comparison between the data dependent and data independent approaches.

However, recently there has been a lot of work in scaling kernel methods to massive amounts of data with inspiration from neural networks. Neural networks scale very well because they can be trained using gradient descent by minimizing the loss wrt the parameters in minibatches. So there is never the full set of data stored within the machine at a single time. [Ma and Belkin, 2017] shows examples using SGD and minibatches with regression and classification that scales to millions of points. Similar results were achieved with sparse approximations for variational GPs [Snelson and Ghahramani, 2007, Bauer et al., 2016] trained with stochastic gradient descent. The most scalable SOTA can be seen from [Meanti et al., 2020] as they pushed the boundary even further with combination of clever software architecture and a conjugate gradient-based scheme called Falkon that scaled regression and classification tasks to billions of points. This feat is also reached for exact GPs as well using matrix-vector-multiplication (MVM) [Gardner et al., 2018] and stochastic variational inference [Matthews et al., 2017]. It's safe to say that kernel methods are now scalable.

Software

Aside from their flexibility and scalability, the success of neural networks's can be attributed to the amount of quality software tools available from many universities and companies. Libraries such as JAX [Bradbury et al., 2018], tensorflow [Abadi et al., 2016], pytorch [Paszke et al., 2019], mxnet Chen et al. [2015] and chainer [Tokui et al., 2019] are currently the most popular libraries which feature easy tensor manipulations and operations on CPU, GPU and TPU [Wang et al., 2019b]. Even the most popular generalized library in the world (scikit-learn, Varoquaux et al. [2015]) has a set of neural network functionality. As alluded to in the introduction, author implementations are good but this doesn't aid reproducibility outside of a few researchers. Well-tested libraries provide more general use cases with more collaborative effort to absorb all of the improvements and optimizations into a well defined framework.

The software landscape for kernel methods is different. There were a few libraries in python and they are not very popular within the community. To cite a few key packages: [Varoquaux et al., 2015] has a fairly complete package for kernel methods functionality and [Raamana, 2020, Lauriola and Aiolli, 2020] have test suites for multiple kernel learning with interoperability with scikit-learn. In terms of scalability, recently [Charlier et al., 2020] is a CUDA-based library which feature "lazy tensors" which compute the kernel matrices in batches to prevent memory overflow. This is a general library with API available for python, R and MATLAB. Their implementation helped scale both the Falkon [FAL, 2020] and MVM [Gardner et al., 2018] frameworks to billions of data points. Julia [Bezanson et al., 2017] is a relatively new language but it already has many libraries available and it attempts to provide cohesion between them. Overall, it appears that the landscape is changing and we hope there will be more interest in the future.

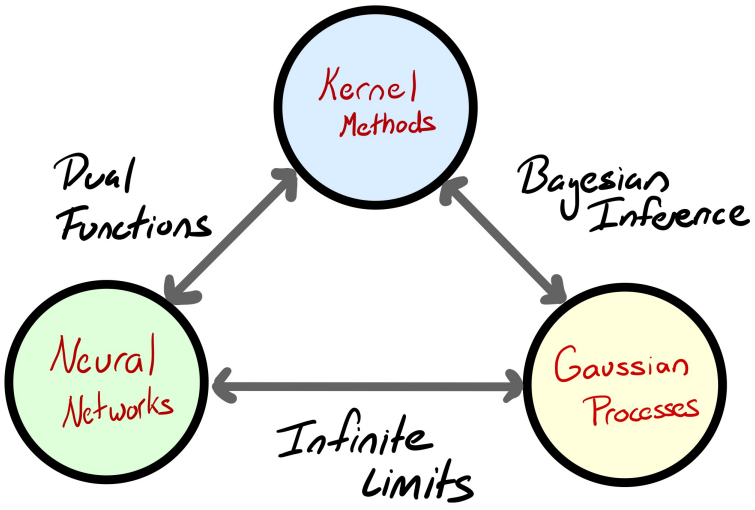


Figure 2.3.: This showcases the connections between kernel methods, Gaussian processes and neural networks. 1) Gaussian process methods are a Bayesian interpretation of many kernel methods, 2) Neural networks are a composition of simple basis function in the primal space where as kernel methods are within the dual space, and 3) if we take the infinite limit of a neural network, we get a Gaussian process. Many times the differences are in the inference procedure when we find the best parameters θ^*.

Dualities to Other Methods

The biggest case for studying kernel methods in general is the fact that there is a deep connection between kernel methods and other methods such as Gaussian processes and neural networks. All of these methods are considered black boxes to some extent; for example in the case of kernel methods, they form implicit maps which are not accessible and for neural networks, they can have millions of parameters which are uninterpretable. So by studying and opening the black-box on one set of methods can unveil insights within another. Neural Networks have been shown in theory and practice that they can approximate any function to a degree of accuracy where given enough layers and enough width, they can approximate any function arbitrarily well. A 2-layer neural network can be represented by a kernel function by its dual representation [Bishop, 2007] which bridges a connection between the two approaches to solving the same problem. [Belkin et al., 2018b] made the case that one needs to understand kernel learning to understand deep learning. For example, they showed the existence of the double descent curve [Belkin et al., 2018a, 2020] via kernel methods. In tandem, another branch emerged as a subfield: the class of kernel functions called neural tangent kernels (NTK) [Jacot et al., 2018, Li and

Liang, 2018] where are infinitely-width neural networks. This lead to an entire new class of algorithms which acts as a trainable infinite-layer neural network. This has allowed researchers to study many neural network architectures through the lens of kernel methods including CNNs [Novak et al., 2019], Transformers [Hron et al., 2020], and Bayesian Deep Ensembles [He et al., 2020] as well as many pathologies like trainability and generalization [Xiao et al., 2019, Belkin et al., 2018b] and information [Shwartz-Ziv and Alemi, 2019]. And just like the neural network community, the software landscape has grown as well within just the last few years [Novak et al., 2020].

The same is found for the connection of kernel methods to Gaussian processes. For example, kernel methods for regression and classification have a connection to Gaussian processes [Murphy, 2012, Kanagawa et al., 2018a]. The kernel ridge regression can be seen as a regularized maximum a posteriori (MAP) estimate of f as we only find a single set of parameters. GPs use this same formulation to specify a prior function parameterized by a mean and covariance function. After maximizing the marginal likelihood function via optimization, they use conditioning to obtain a predictive mean and variance function [Bishop, 2007]. [Kanagawa et al., 2018b] show that the predictive variance can also be used for the Kernel ridge regression algorithm to serve as an upper bound to the GP predictive variance due to the differences in the training procedure. GPs can also be derived from infinite neural networks with a Gaussian prior over the weights [Neal, 1996]. Which now comes full circle between Kernel methods, neural networks and Gaussian processes (figure 2.3). It it worth noting, that the NTK mentioned above has also shed many insights into the parallels between neural networks and Gaussian processes [Lee et al., 2018, de G. Matthews et al., 2018b, Novak et al., 2019, Yang, 2019] as a bridge between the two methods. All of this is to say that there are connections that exist between kernel methods and other popular machine learning methods like deep learning and Gaussian processes. So it is worth investigating kernel methods as there are still potential ideas that can be transferred between the methods.

2.3. The Paper

Title
Kernel methods and their derivatives: Concept and perspectives for the earth system sciences [Johnson et al., 2020d]

Authors
J. Emmanuel Johnson, Valero Laparra, Adrian Perez-Suay, Miguel D. Mahecha, Gustau Camps-Valls

2.3.1. Summary

In this paper, we address the solution proposed in section 1.4.1. We provide baseline examples of how one can use sensitivity analysis for kernel methods to give insight into how the inputs affect modeled outputs. We do an extensive literature review related to kernel methods for modeling, data analysis and exploration. We give a formal formulation of how one can take the derivative of any kernel function. We subsequently follow this with the formulation for how one can take the gradient for most kernel methods; we include regression with kernel ridge regression (KRR) [Schölkopf and Smola, 2002a], classification with Support vector machines (SVMs) [Schölkopf and Smola, 2002a], density estimation with kernel entropy components (KECA) [Jenssen, 2010, Izquierdo-Verdiguier et al., 2017] and dependence estimation with the Hilbert-Schmidt Independence Criterion (HSIC) [Gretton et al., 2005]. We relate this to sensitivity and give toy examples for each of the kernel methods. We finish the paper with proof-of-concept examples on real Earth system data for each of the aforementioned kernel methods with appropriate experiments to justify the use of kernel methods for modeling coupled with sensitivity measures for interpretable and exploratory data analysis. Some examples include spatio-temporal smoothing using KRR, drought predictions using SVMs, density ridge visualization from probability density estimates of spatial regions using KECA, and variable dependencies with seasonal changes using HSIC.

2.3.2. Contributions

This paper is one of the first all-inclusive look at derivatives for kernel methods as a whole. In the paper we acknowledged all of the preliminary works where people had applied derivatives of kernel methods for various applications but we also fill in a crucial gap within the literature: never has the formulation for the derivatives of kernel methods in general been laid out in a step-by-step fashion motivated by the conceptual and applied examples. Not only did we summarize the current work available, we also gave motivation and explanations using toy examples and use cases as to *why* it works. We paid special attention to sensitivity analysis as it is a crucial component for explainability for machine learning models and showcased how it can be applied and interpreted in non-parametric models. We supported all interpretations and intuitions with motivating toy examples and realistic applied examples. The examples we gave for Earth system data catered to how we could use kernel methods for data analysis and understanding and not just for modeling. For example, we used the combination of regression and sensitivity analysis to analyze the trade-off between spatial-temporal feature representations. While our original intention was analyzing derivatives for kernel functions in discriminant models like for regression and classification, we recognized that the first principles of derivatives of kernel functions is in fact generalizeable across other models like density estimation and independence estimation. One one hand, we hope that our work serves as a stepping stone for other more ambitious and targeted applications of kernel methods for EO applications. On the other hand, we hope that we give practitioners more con-

fidence that their kernel-based algorithms can be effectively analyzed in the presence of complex multivariate EO data.

2.3.3. Reproducibility

Code Base

All of the available code for the paper is available online[1]. This includes all of the toy experiments and the real experiments for the regression, classification and dependence estimation. This is the main repo for the majority of the paper which was done in Python. The density estimation experiments were done by Valero Laparra (in MATLAB) and are available online.

Datasets

For the datasets used in the applied experiments, we use the Earth System Data Cubes [Mahecha et al., 2020] which are available from the Earth System Data Lab[2].

Software

A didactic kernel methods library, called *jaxkern*[3], was developed in order to facilitate reproducibility and to promote future experiments. It is based on *JAX* [Bradbury et al., 2018] which allows for easy parallization, easy JIT compilation, and one can use the same code for CPUs, GPUs and TPUs. It also uses Objax [Objax Developers, 2020] as a simplified Object-Oriented Programming (OOP) abstraction to provide an easier interface to JAX for users. It hopes to fill the void within the kernel methods community as a new library for good quality, reproducible kernel methods.

2.4. Further Research Directions

Since the publication of the journal article listed in the previous section, we reflected upon the limitations of the work as well as outlined future opportunities for further research. In the remainder of the chapter, we will highlight some of the pitfalls as well as future solutions.

2.4.1. Limitations

Unsupervised Kernel Methods

One limitation of this work is unexplored sensitivity of the hyper-parameters for the unsupervised kernel methods. Kernel methods are very flexible but their hyper-parameters

[1] https://github.com/IPL-UV/sakame
[2] https://www.earthsystemdatalab.net/
[3] https://github.com/IPL-UV/jaxkern

can impact the model results. Methods like regression or classification minimize a cost function to find the best parameters which we demonstrated for the Kernel Ridge Regression algorithm as well as the Support Vector classification scheme. However, there is no community standard for finding the hyperparameters of unsupervised kernel methods like density estimation and Hilbert-Schmidt Independence Criterion besides the standard ad-hoc methods. While we believe derivatives and sensitivity analysis can help shed light into the unsupervised learning regimes, this is still a relatively new area to be explored with significantly different objectives. One potential application would be to use sensitivity analysis to assess how these chosen parameter values assess. The user should have some intuition about which kernel function is appropriate for their application so SA could an additional tool to give insight into the best kernel parameters. This is potentially a very powerful combination as it gives the user access to impose inductive biases (via kernels) as well as post-hoc analysis (via sensitivity analysis).

Lack of Complex Kernel Functions

In this work, we did not use any other kernel function besides the standard RBF kernel function. This is a very smooth kernel which is infinitely differentiable so it is a very good candidate for exploring derivatives. However, we did not assess how our derivative-based applications work for more complex and expressive kernel functions. Model misspecification is a problem in machine learning [Dennis et al., 2019] as one should impose relevant inductive biases to achieve the most optimal results. At the very least, one should check how these methods perform for more expressive kernel solutions, e.g. the Automatic-Relevance-Determination (ARD) kernel, the Matérn kernel or even the Neural Network kernel [Rasmussen and Williams, 2006]. In theory, as long as the kernel is differentiable, one could apply derivative-based analysis such as sensitivity analysis but for more complex and non-smooth solutions, it may be harder to interpret the sensitivity of the learned features due to the presence of many gradients due to non-smooth solutions. This problem surfaced in the XAI community [Shrikumar et al., 2017] which resulted in augmented methods to combat this. This is a problem that could also arise with multiple-kernel learning with complex kernel functions which is a very important field within the kernel methods literature.

2.5. Concluding Remarks

More recently, the explainable AI (XAI) community have been promoting a related method within the context of machine learning: saliency maps (a.k.a. feature attribution or heatmaps). While under a different name, these methods are similar and also seek to find the influential parts of the model which are the most relevant for the model's predictions. This is predominant in the neural networks community (especially for classification problems). The gradients of neural networks are available through automatic differentiation so this is a natural choice to try and use gradient-based techniques. However, the

standard gradient-based analysis methods [Baehrens et al., 2010, Simonyan et al., 2014] have problems such as gradient saturation which lead to maps which were visually noisy and difficult to interpret [Shrikumar et al., 2017]. Other methods try to overcome these limitations by augmenting the standard gradient with noise, i.e. SmoothGrad [Smilkov et al., 2017], by summing interpolated gradients (i.e. Integrated Gradients [Sundararajan et al., 2017]), or by multiplying the gradients with the original input (i.e. Gradient-Input [Shrikumar et al., 2017, Ancona et al., 2018]). Other approaches have tried to modify the back-propagation to comply better with the gradients, e.g. Layer Relevance Propagation [Binder et al., 2016]. For a full review of the XAI methods, see [Arrieta et al., 2020, Payrovnaziri et al., 2020]. While we mentioned that kernel methods have deep connections to neural networks, some of these methods and augmentations could potentially be useful especial with more complex kernel functions.

Spatial-Temporal Earth science data is very high-dimensional and very large scale. In order for kernel methods to compete with other methods, one will need to investigate how each of the approximation methods work with sensitivity analysis. Matrix approximations could lead to a more difficult, less intuitive formulation which needs to be explored in greater detail. SA is not only useful in the setting as future work to explore exploration, SA could also help us to understand the trade-off between the number of features/data points needed to capture the relevant features for the kernel approximation. Further work is needed for this direction but it is a very important component when applying these kernel methods for real large-scale datasets.

In this work, we focused on derivative-based sensitivity measures. However, as alluded to in the introduction, there is a rich history of methods like variance-based schemes such as Sobol indices [Sobolá, 2001]. Often users choose their sensitivity analysis measure based on the computational resources available or their legacy. It would be good to bridge the gap across methods and find commonalities between them so that we can give users more options to question and improve their models. It would be good to compare how other SA methods can be applied and how these results compare to derivative-based methods. In the regression and classification problems, these are prime candidates to experiment with other metrics such as Sobol indices or even polynomial chaos regimes. However, in the unsupervised kernel methods, it is not immediately clear if this is feasible or even possible.

We have demonstrated derivative-based analysis methods for kernel methods in the context of EO data [Johnson et al., 2020d]. Not only did we give intuitive examples of how each kernel method can make use of its derivative to give insight, but we have also given proof-of-concept examples of how they can be used in challenging applications with spatial-temporal Earth datasets. In the previous section we also gave an overview of the pitfalls and limitations, and suggested concrete solutions to overcome these. Because SA is a very flexible framework, we anticipate a wider use of sensitivity analysis in kernel methods in particular. This paper presented and formalized the field and showcased illustrative examples in both synthetic and real life problems in the Earth sciences.

3. Uncertain Inputs in Gaussian Processes

Contents

Gaussian processes (GPs) are considered the golden standard for modeling uncertainty and obtaining viable confidence intervals for the output estimates. However, one aspect that is often overlooked in the applied GP literature is the incorporation of input uncertainty. In this chapter, we examine the GP literature for improving the confidence intervals of GPs. The included journal article examines a real-world example of the viability of one method in practice. Since only one method is demonstrated in the journal article, subsequent experiments are provided to further demonstrated the viability of the method. This chapter concludes with a detailed overview of the strengths and limitations of the work as well as some future directions.

3.1. Uncertain Inputs

In real-world applications, we often need to consider uncertain inputs in our machine learning models. Every instrument we use to collect data will have some level of uncertainty and this is often explicitly available in certain datasets like observational products [Mahecha et al., 2020] and satellite datasets [Chalon et al., 2001, Blumstein et al., 2004, Camps-Valls et al., 2012]. Alternatively, we could have inputs from other models like a trained regressor which would also have uncertainty. Not including this information into our datasets can have adverse effects on our predictions and uncertainty as we are not actually propagating error through our model. So we should definitely take this information into consideration when choosing the appropriate model. In this first section, we will set up the problem and cover some of the methods found in the literature.

3.1.1. Gaussian Process Predictions

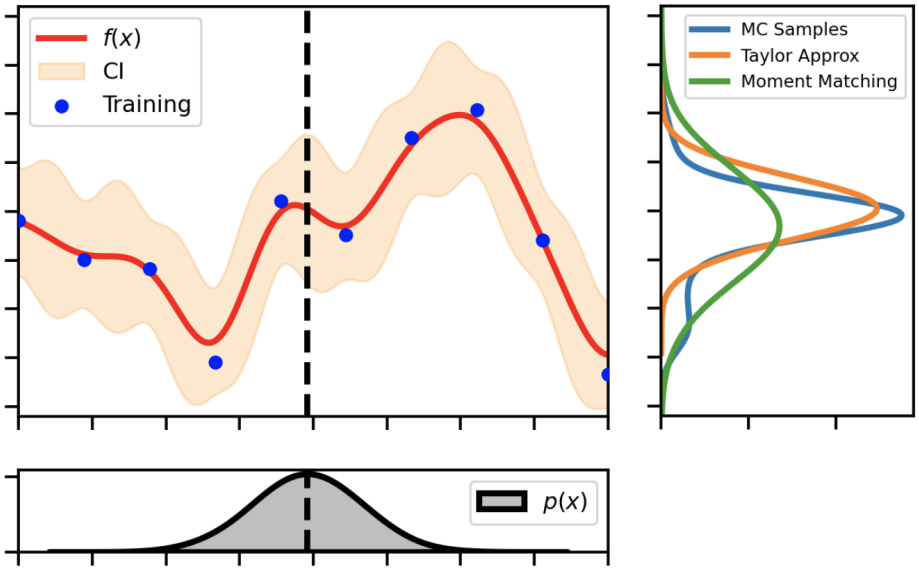


Figure 3.1.: A demonstration showing how an uncertain input propagates through an already fitted GP and the resulting distributions. In this example, we have an already fitted GP function, $\mathbf{f}$ to a dataset $\mathbf{X}, y$ which provides us with a predictive mean and variance function. However, we showcase that a sample $\mathbf{x} \sim p(\mathbf{x})$ propagated through the predictive mean function, and we see that the output distribution is non-Gaussian. To showcase the output distribution, we show how this looks using different methods: 1) $1K$ Monte Carlo samples, 2) the linearized Taylor Series approximation, and 3) the Gaussian approximation using the Moment-Matching method.

In Gaussian processes, the original formulation dictates that we assume there is some noise in the observations, y and that we observe the real inputs $\mathbf{x}$. So we'll see that this it is not trivial to modify this formulation to account for uncertain inputs. Let's assume that we have a data set $\mathcal{D} = \{\mathbf{X}, \boldsymbol{y}\}$. In this case we assume the following relationship between our inputs, $\mathbf{x}$, and outputs, y:

$$y_i = \mathbf{f}(\mathbf{x}_i) + \epsilon_y \tag{3.1}$$
$$\epsilon_y \sim \mathcal{N}(0, \boldsymbol{\Sigma}_y^2) \tag{3.2}$$

Let's also assume that we have a standard GP model optimized and fitted to this data set. We're not assuming noisy inputs during the training phase so we will use the standard log-likelihood maximization procedure. However, during the testing phase, we will assume that our inputs are noisy. For simplicity, we can assume our test data set is normally distributed with a mean $\boldsymbol{\mu}_{\mathbf{x}}$ and variance $\boldsymbol{\Sigma}_{\mathbf{x}}$. So we will have:

$$\mathbf{x}_* \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) \tag{3.3}$$

or equivalently we can reparameterize it like so:

$$\mathbf{x}_* = \boldsymbol{\mu}_{\mathbf{x}} + \epsilon_{\mathbf{x}} \tag{3.4}$$
$$\epsilon_{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{x}}) \tag{3.5}$$

If we consider the predictive distribution given by $p(y_*|\mathbf{x}_*, \mathcal{D})$, we need to marginalize out the input distribution. So the full integral appears as follows.

$$p(\mathbf{f}_*|\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}, \mathcal{D}) = \int p(\mathbf{f}_*|\mathbf{x}_*, \mathcal{D})\, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})\, d\mathbf{x}_* \tag{3.6}$$

If we use the GP formulation, we have a closed-form deterministic predictive distribution for $p(\mathbf{f}_*|\mathbf{x}_*, \mathcal{D})$. Plugging this into the above equation gives us:

$$p(\mathbf{f}_*|\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}, \mathcal{D}) = \int \mathcal{N}\left(\mathbf{f}_*|\boldsymbol{\mu}_{\mathcal{GP}}(\mathbf{x}_*), \boldsymbol{\Sigma}^2_{\mathcal{GP}}(\mathbf{x}_*)\right)\, \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})\, d\mathbf{x}_* \tag{3.7}$$

So this integral is intractable because if we consider the terms within the GP predictive mean and predictive variance, we will need to calculate the integral of an inverse kernel function, $\mathbf{K}_{\mathcal{GP}}^{-1}$. Below we outline some of the most popular methods found in the literature.

3.1.2. Monte Carlo Sampling

The most exact solution would be to use Monte-Carlo (MC) simulations [Marzjarani, 2019]. We draw T samples from the distribution of our $\mathbf{x} \sim \mathcal{N}(\mathbf{x}_*|\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})$ and propagate

this through the predictive mean and standard deviation of the Gaussian process to obtain our samples:

$$p(\mathbf{f}_*|\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}, \mathcal{D}) \approx \frac{1}{T}\sum_{t=1}^{T} \mathcal{N}\left(\mathbf{f}_*|\boldsymbol{\mu}_{\mathcal{GP}}(\mathbf{x}_*^t), \boldsymbol{\Sigma}^2_{\mathcal{GP}}(\mathbf{x}_*^t)\right). \tag{3.8}$$

This operation will be exact as the number of MC samples, T grows. In addition, one could use any distribution they want to represent the inputs $\mathbf{x}$ like the T-Student for more noisy scenarios. The downside is that this method can be very expensive as we will have to propagate our inputs through the predictive mean function T times. This is especially true for the exact GP but possibly can be mitigated by more sparse approximations [Bui et al., 2017b] or faster sampling schemes [Wilson et al., 2020, Pleiss et al., 2020]. This method has not been demonstrated in real world problems, only in toy examples [Girard et al., 2002b]. There have been many developments in the literature with regards to MC methods for inference including Gibbs sampling [Titsias et al., 2008], Elliptical slice sampling [Murray and Adams, 2010], and No U-Turn Sampler [Phan et al., 2019]. However, none of these methods have been used for error propagation. MC methods have gotten more efficient over the years [Phan et al., 2019, Lao et al., 2020] so this method has the potential to be critical in applications with high uncertainty and a more thorough investigation of the parameters is needed especially with small-medium data problems.

3.1.3. Gaussian Approximation

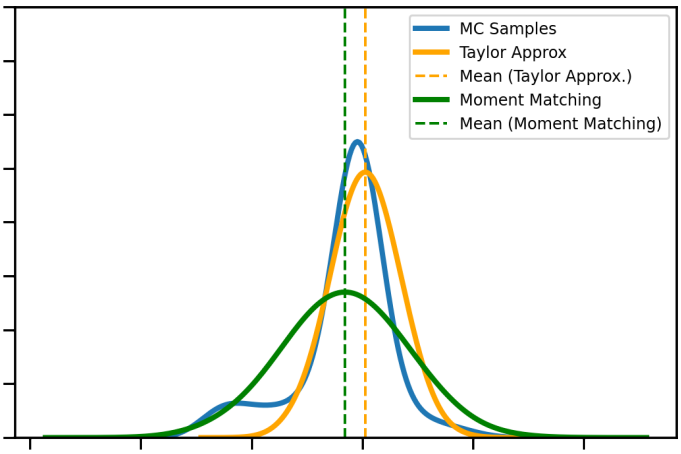


Figure 3.2.: A closer look at the shape of the posteriors for each of the uncertain operations (Taylor's approximation, Moment Matching) versus the golden standard Monte Carlo sampling. The Taylor's approximation is a *linearization* and it approximates the mode of the output distribution whereas the Moment matching approximates the mean and covers the entire space.

The integral of the GP predictive distribution is intractable as mentioned before so we need a way to approximate this distribution. In this family of methods, we approximate the predictive distribution as a Gaussian with the first and second moments. We can compute the moments of the predictive mean and variance equations by using the law of iterative expectations [Endou, 2019]:

$$m(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) = \mathbb{E}_{\mathbf{x}_*}\left[\mathbb{E}_{f_*}\left[f_*|\mathbf{x}_*\right]\right] = \mathbb{E}_{\mathbf{x}_*}\left[\boldsymbol{\mu}(\mathbf{x})\right] \tag{3.9}$$

$$\begin{aligned} v(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) &= \mathbb{E}_{\mathbf{x}_*}\left[\mathbb{V}_{f_*}\left[f_*|\mathbf{x}_*\right]\right] + \mathbb{V}_{\mathbf{x}_*}\left[\mathbb{E}_{f_*}\left[f_*|\mathbf{x}_*\right]\right] \\ &= \mathbb{E}_{\mathbf{x}_*}\left[\boldsymbol{\Sigma}^2(\mathbf{x}_*)\right] + \mathbb{V}\left[\boldsymbol{\mu}(\mathbf{x}_*)\right] \\ &= \mathbb{E}_{\mathbf{x}_*}\left[\boldsymbol{\Sigma}^2(\mathbf{x}_*)\right] + \mathbb{E}_{\mathbf{x}_*}\left[\boldsymbol{\mu}^2(\mathbf{x}_*)\right] - \mathbb{E}^2_{\mathbf{x}_*}\left[\boldsymbol{\mu}(\mathbf{x}_*)\right] \end{aligned} \tag{3.10}$$

So our final set of equations involve expectations over varying degrees of the predictive mean and variance equations for the GP algorithm. There are two competing methods in the literature for computing the expectations and variances of the predictive mean and variance: linearization and moment-matching. Linearization entails approximating the expectation with a Taylor's expansion and moment-matching entails computing the moments exactly and then approximating the remaining integrals with quadrature methods like Gauss-Hermite or unscented transformations. The Taylor's transformation is easier to compute but less exact whereas the moment matching method is more exact but more expensive to compute. In paper 2 [Johnson et al., 2020a], we chose the linearization approach yet we will outline below the details of both approaches as well as some other approaches.

Taylor's Approximation

This is the simplest approach that is found in many of the earlier uncertain input GP literature. In this framework, we approximate the expected predictive mean and variance via a first and second order Taylor. Using this expansion, it is easier to compute the first and second moments (mean and variance) of the predictive distribution. This is a relatively fast and approximate method which incorporates the uncertain inputs information into the predictive variance without needing to retrain the GP model. The equations are summarized below:

$$\tilde{\boldsymbol{\mu}}_{\text{LinGP}}(\mathbf{x}_*) = \boldsymbol{\mu}_{\text{GP}}(\boldsymbol{\mu}_{\mathbf{x}_*}) + \underbrace{\frac{1}{2}\text{Tr}\left\{\frac{\partial^2 \boldsymbol{\mu}_{\text{GP}}(\boldsymbol{\mu}_{\mathbf{x}_*})}{\partial \mathbf{x}_* \partial \mathbf{x}_*^\top}\boldsymbol{\Sigma}_{\mathbf{x}_*}\right\}}_{\text{second Order}} \tag{3.11}$$

$$\tilde{\boldsymbol{\Sigma}}^2_{\text{LinGP}}(\mathbf{x}_*) = \boldsymbol{\Sigma}^2_{\text{GP}}(\boldsymbol{\mu}_{\mathbf{x}_*}) + \underbrace{\frac{\partial \boldsymbol{\mu}_{\text{GP}}(\boldsymbol{\mu}_{\mathbf{x}_*})}{\partial \mathbf{x}_*}^\top \boldsymbol{\Sigma}_{\mathbf{x}_*} \frac{\partial \boldsymbol{\mu}_{\text{GP}}(\boldsymbol{\mu}_{\mathbf{x}_*})}{\partial \mathbf{x}_*}}_{\text{1st Order}} + \underbrace{\frac{1}{2}\text{Tr}\left\{\frac{\partial^2 \boldsymbol{\Sigma}^2_{\text{GP}}(\boldsymbol{\mu}_{\mathbf{x}_*})}{\partial \mathbf{x}_* \partial \mathbf{x}_*^\top}\boldsymbol{\Sigma}_{\mathbf{x}_*}\right\}}_{\text{second Order}} \tag{3.12}$$

As shown, we still include the original predictive mean and variance terms (see [Girard et al., 2002b, Bijl, 2018] for the full derivation or appendix B.4 for a summarized

derivation). In the end, this approximation augments the predictive mean and variance equations with terms that incorporate the first derivative of the predictive mean (1st order) and the second derivative of the predictive mean and variance (second order). This was originally proposed in [Girard et al., 2002b, Girard and Murray-Smith, 2003, Girard, 2004] by augmenting the Gaussian process predictive mean and variance with the derivative of the predictive mean and the trace of the predictive variance. Subsequently, we saw other approaches implement the same strategy with great success on dynamical problems [Oakley and O'Hagan, 2002, Oakley and OHagan, 2004].

In [McHutchon and Rasmussen, 2011, McHutchon, 2014]m they modeled this during the training regime as well by incorporating the linearization term into the GP likelihood. The results were promising and the confidence intervals were better. However, due to including the derivative of the kernel in the formulation, this resulted in a cyclic optimization scheme where one would need to optimize, find the derivative, repeat until convergence which is an expensive operation. In general, the dynamic GP community advises against using the Taylor's approximation scheme because the variance estimates are unreliable after many iterations [Deisenroth and Mohamed, 2012]. However, they do acknowledge that this method is a good alternative when one needs an easy and scalable approximation and it was revisited as an alternative in GP classification [Villacampa-Calvo et al., 2020].

Moment Matching

Moment matching is one of the most commonly used methods to date for dealing with uncertain predictions in GPs [Deisenroth and Mohamed, 2012]. It works by computing the first and second moments of the new predictive distribution and then applying quadrature methods to solve all of the remaining integrals. So explicitly we need to take expectations (integrals) of the GP predictive mean and variance w.r.t. our distribution for $\mathbf{x}_*$:

$$\tilde{\boldsymbol{\mu}}_{MMGP}(\mathbf{x}_*) = \int \boldsymbol{\mu}_{GP}(\mathbf{x}_*) p(\mathbf{x}_*) d\mathbf{x}_* \tag{3.13}$$

$$\tilde{\boldsymbol{\Sigma}}^2_{MMGP}(\mathbf{x}_*) = \int \boldsymbol{\Sigma}^2_{GP}(\mathbf{x}_*) p(\mathbf{x}_*) d\mathbf{x}_* + \int \boldsymbol{\mu}^2_{GP}(\mathbf{x}_*) p(\mathbf{x}_*) d\mathbf{x}_* - \left[\int \boldsymbol{\mu}_{GP}(\mathbf{x}_*) p(\mathbf{x}_*) d\mathbf{x}_* \right]^2$$

After some manipulation, this results in the following equations for the predictive mean and variance:

$$\tilde{\boldsymbol{\mu}}_{MMGP}(\mathbf{x}_*) = \Psi_1^\top \boldsymbol{\alpha} \tag{3.14}$$

$$\tilde{\boldsymbol{\Sigma}}^2_{MMGP}(\mathbf{x}_*) = \psi_0 - \operatorname{Tr}\left(\left(\mathbf{K}_{GP}^{-1} - \alpha\alpha^\top \right) \Psi_2 \right) - \operatorname{Tr}\left(\Psi_1 \Psi_1^\top \alpha\alpha^\top \right), \tag{3.15}$$

where we have $\mathbf{\Psi}_i$ quantities called kernel expectations denoted by:

$$[\psi_0]_i = \int k(\mathbf{x}_i, \mathbf{x}_i) p(\mathbf{x}_i) d\mathbf{x}_i \tag{3.16}$$

$$[\Psi_1]_{ij} = \int k(\mathbf{x}_i, \mathbf{y}_j) p(\mathbf{x}_i) d\mathbf{x}_i \tag{3.17}$$

$$[\Psi_2]_{ijk} = \int k(\mathbf{x}_i, \mathbf{y}_j) k(\mathbf{x}_i, \mathbf{z}_k) d\mathbf{x}_i. \tag{3.18}$$

where we still include the original predictive mean and variance terms, see [Dutordoir, 2016, Bijl, 2018, Deisenroth, 2010] for the full derivation or appendix B.5 for a summarized derivation. This is a much more complex set of equations than that of the Taylor's approximation. What was reduced to derivatives in equations 3.11, this formulation still has integrals; in particular expectations over kernel functions. So we need to use quadrature methods to calculate these quantities. Compared to the linearization approach, this will give us a more exact solution and better representation than the Taylor approximation especially for more complex functions. This method was used in [Girard et al., 2002b, Candela, 2004] for dynamical systems problems and later it become more popular in applications such as the PILCO problem where [Deisenroth, 2010] used the same formulation. Later, this method was seen in more recent GP developments like the Bayesian GPLVM [Titsias and Lawrence, 2010] and the original deep GP [Damianou et al., 2014, Damianou, 2015] where they use the variational approach.

This is often the preferred method for many applications with uncertain inputs. One advantage is the geometric meaning as it is akin to approximating the forward Kullback-Leibler Divergence (KL-Divergence) between a prior p and an approximate variational distribution q, i.e. $\mathrm{D}_{\mathrm{KL}}[p||q]$, the KL-Divergence. The moment matching distribution is similar to the approximate variation distribution q and the uncertain input data is similar to the prior term p [Deisenroth, 2010]. The forward KL-Divergence is a conservative estimate to ensure all regions of $p(x) > 0$ are covered by $q(x)$. This is very similar to the approach taken by the α-divergence and expectation propagation when $\alpha = 1$ [Bui et al., 2017a]. However practically, this is an expensive measure to calculate due to the kernel expectations. It is only *exact* for specific kernel functions that have been derived like the linear, RBF [Girard et al., 2002b, McHutchon, 2014, Damianou et al., 2014] and spectral kernels [Dutordoir, 2016]. In all other cases, the integrals need to be approximated via quadrature methods. The Gauss-Hermite is the most popular method found in standard GP toolboxes [Matthews et al., 2017, Gardner et al., 2018] but there have been explorations to use unscented transform [de Souza et al., 2019] which are more scalable and are exact enough in lower dimensional settings. But in practice, Monte-Carlo sampling is the preferred method of choice for dealing with this in stochastic optimization settings for types of variational GPs [Dutordoir et al., 2018].

3.1.4. Toy Example

Let us show in the following toy example how each of these methods perform and how well they improve the confidence intervals of the data. More specifically, we will have a qualitative assessment of the confidence intervals and see if they sufficiently capture the outliers within the data. In Figure 3.3 have a standard GP and we see that the confidence intervals do not sufficiently encapsulate the noisy inputs. The figure showcases a simple "nearly-square" sine wave and how four different GP formulations approximate the variance of the function. The standard GP in figure 3.3 clearly does not reflect the errors in the inputs as there are still many points that lie outside of the confidence regions especially around the gradients.

All of the augmented GPs (i.e. Monte Carlo, Taylor approximation, moment-matching) showcase better predictive variance estimates. The Monte Carlo method in figure 3.3 showcases the most exact solution as we see that the only regions where the predictive variance is higher is near the gradients. However, it is overconfident in the regions where the function is plateaued. The Taylor expansions (1st and second order) both have better confidence intervals along the gradients and the plateau regions. They appear to be less confident and overestimate the uncertainty compared to the Monte Carlo approximation. The moment matching approximation has similar confidence intervals to the Taylor's approximations and are a lot smoother. It is important to note that the mean predictions for all of the methods are similar but we get better confidence intervals in all of the proposed methods above.

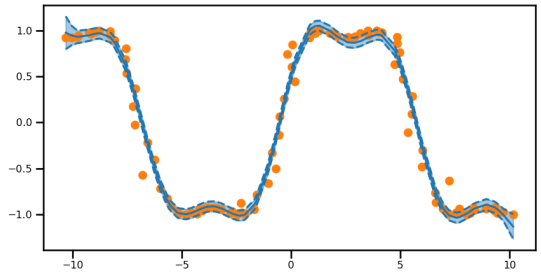
3.2. The Paper

Title
Accounting for Input Noise in Gaussian Process Parameter Retrieval [Johnson et al., 2020a]

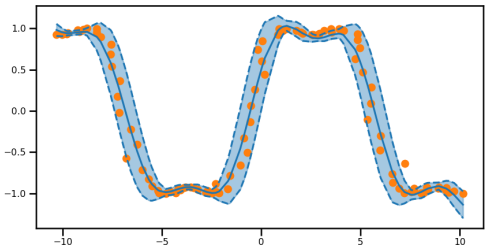
Authors
J. Emmanuel Johnson, Valero Laparra, Gustau Camps-Valls

3.2.1. Summary

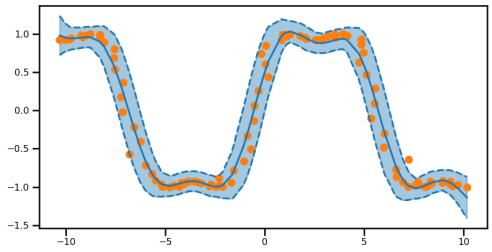
In this paper, we address the solution proposed in section 1.4.2. We show how one can propagate input uncertainty through the standard Gaussian process model. For this work, we used the first-order Taylor series approximation which only affects the predictive variance. It assumes that the variance is proportional to the square of the derivative scaled by the covariance of the input data. In other words, we augment the predictive variance by including derivative information of the predictive mean function. Using a toy example,



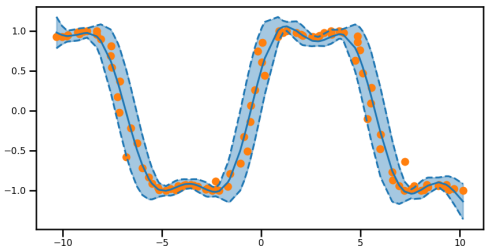
(a) Standard GP Predictions



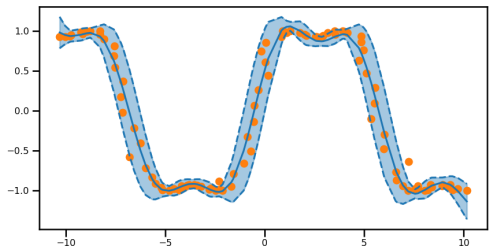
(b) Monte Carlo Sampling



(c) Moment Matching



(d) Taylor Expansions (1st Order)



(e) Taylor Expansions (2nd Order)

Figure 3.3.: A uni-dimensional toy example showing the standard Gaussian process algorithm and how well a standard GP predicts the mean and variance. For this example, an input noise of $\boldsymbol{\Sigma}_{\mathbf{x}} = 0.3$ and an output noise $\boldsymbol{\Sigma}_{\mathbf{y}} = 0.05$ was used for this demonstration. We see the standard GP predictions do not encapsulate all of the outliers. We showcase a) the assumed correct solution using 10000 Monte Carlo samples, b) the moment matching approximation, c) the linearization approximation using the 1st order Taylor expansion, d) the linearization approximation using the second order Taylor expansions.

we showed how this is superior to the standard Gaussian process as the noisy simulated data was more likely to be inside the confidence intervals. Using real high-dimensional spectral reflectances, we used a GP to predict global temperature. We showed that the

predictive variance for this solution was visually worse for the standard GP than our improved augmentation. We also showed that the augmented GP provided beter calibrated predictive variance measures than the standard GP.

3.2.2. Contributions

Gaussian processes are one of the most widely used algorithms in remote sensing and Earth observation applications [Camps-Valls et al., 2016]. However, standard methods generally do not take into account input uncertainty. This paper is one of the first available studies where we actively solve the issue of input error propagation for GP models within large scale, multi-dimensional datasets. There are currently a lot of methods available for input error propagation within GPs but they have not been widely used in practice. In addition, there are not many studies where they really assess the impact of input error on the confidence intervals of GPs. Furthermore, we demonstrate that even a simple, relatively inexpensive, and intuitive modification to the GP algorithm results in improvements. The results from this work have implications in many popular applications in EO including kriging, time series analysis and surrogate modeling. We have successfully motivated the community to take into account input uncertainty as it really can impact the model results and subsequent decisions made by users.

3.2.3. Reproducibility

Code Base

All of the available code for the paper is available at the following url[1]. This includes all of the toy experiments and the real experiments for the IASI dataset. This is the main repository for the majority of the paper which was done in Python.

Datasets

Unfortunately, this dataset is not available for the public. We chose it as a representative dataset which has been a great testbed for many ML methods and applications [Blumstein et al., 2004, Camps-Valls et al., 2012, Hilton et al., 01 Mar. 2012]. We highly encourage new researchers to choose datasets that are freely available especially given the fact that uncertain inputs for GP research is a relatively new field.

Software

There are already excellent GP libraries available, e.g. *GPFlow* [Matthews et al., 2017] and *GPyTorch* [Gardner et al., 2018]. For more information, please see this guide[2]. So fortunately, there is already a large software presence for Gaussian processes. However,

[1] https://github.com/IPL-UV/gp_error_propagation
[2] https://jejjohnson.github.io/gp_model_zoo/software/

these libraries were already pre-packaged and already optimized. This work for uncertain GPs was fairly exploratory and required a more didactic approach. As such, we become a core contributor to the *GPJax* package (found here[3]) whereby we used their foundation to explore uncertain inputs for GPs. Coincidently, it uses the *JAX* and *Objax* backend which was also used in the software for the previous chapter paper.

Literature

One of the issues found was the lack of summary literature regarding uncertain inputs for Gaussian processes. This is fairly normal but we wanted to contribute to the community by gathering all literature related to uncertain inputs. As such a website[4] has been created to help facilitate future research. In addition, another more general GP literature website[5] was created as a way to help the community have more cohesion between papers.

3.3. Further Research Directions

Since the publication of the journal article [Johnson et al., 2020b] listed in the previous section, further research was done on some alternative ways to handle uncertain inputs. As mentioned in the reproducibility section, most of these ideas are documented on the literature review page[6] but we expand some particular ideas of interest in this thesis. In the remainder of this chapter, we will summarize some of the highlights of the literature and give suggestions for further research.

3.3.1. Follow-up Literature Review

In the introduction to this chapter, pertinent to the above paper, we assume that a GP regression model has already been trained on a dataset $\mathcal{D}$ and only the predictive mean and variance take into account the uncertain data. The methods are advantageous because they are simple and we can use the exact GP formulation without resorting approximations. For intended applications like dynamical systems with time series, propagating the error via the dynamical system formulation is reasonable. However, this is strong simplification for the entire dataset which could be noisy and subsequently compromise the validity of the trained GP model. The predictive variance $\mathbf{\Sigma}^2_{\text{GP}}$ is still independent from the input data so one would still need to augment it to incorporate the input uncertainty. But the learned weights and kernel parameters for the GP model could be changed once we incorporate noisy data assumptions into the model. This would require us to modify the GP formulation in more creative ways. I have outlined a few of the key methods one could do so below.

[3] https://github.com/thomaspinder/GPJax
[4] https://github.com/jejjohnson/uncertain_gps/
[5] https://github.com/jejjohnson/gp_model_zoo/
[6] https://jejjohnson.github.io/uncertain_gps/Notes/literature/

Kernel Functions

Another frame of thinking involves modifying the kernel function to allow one to include the error in the input data. Kernel functions are very powerful and if constructed correctly, one could embed uncertain considerations into the chosen kernel function. This is advantageous as it allows one to incorporate the uncertainty into the training of the GP method in addition to the testing without having to use approximate posterior methods (see latent variables 3.3.1). In [Dallaire et al., 2011], the motivation was to modify the length scale of the RBF kernel to account for the covariance within the inputs. The limitations of this method is that it assumes a constant covariance for each of the inputs which is not flexible. Besides, [McHutchon, 2014] found that the length scales describing the RBF kernel function may collapse to the scale of the covariance. Surprisingly, this approach has not been explored more seeing how a common limitation of Gaussian process methods is the expressiveness of the kernel function [Krauth et al., 2017], and so creating a kernel to incorporate the error in the inputs would be a clever way to mitigate this issue. For example, [Moreno et al., 2003] created a specialized kernel based off of the KL-Divergence which works for Gaussian noise inputs. Even though this is not a valid kernel the results showed improvement in the confidence interval predictions. A clever way to mitigate this is to simply pose it as a function on top of a kernel, i.e. $k(f(x_i), f(x_j))$. This is the basis of the Deep kernel learning [Wilson et al., 2016a,b] literature which features fully parameterized neural networks where their outputs are fed into a simple kernel function; essentially a Gaussian process output layer on a neural network. This flexibility allows one to construct the network to learn certain feature representations where we can embed knowledge to deal with uncertain inputs like the noise constrastive prior [Hafner et al., 2019]. It is also motivating to see methods like the SNGP [Liu et al., 2020] which features a similar architecture to the KL-Divergence but is fully Bayesian with priors over the hyperparameters of the model. They found that the confidence intervals obtained included distance awareness to combat out-of-sample uncertainty which essentially means that data points further away from the input distribution were classified as uncertain.

Heteroskedastic Likelihoods

In this approach, the problem is transformed to finding a function, $g_{\boldsymbol{\theta}}$ over the noise-likelihood, i.e. $\epsilon \sim \mathcal{N}(0, g(\mathbf{x}))$ [Kersting et al., 2007, Lázaro-Gredilla and Titsias, 2011, Zhang and Ni, 2020]. Typically, this noise-likelihood is a constant value, σ_y^2 but noise level cannot capture the input dependent noise. Having a functional form: This would enforce the predictive variance to have some dependencies upon the inputs. The challenge is that this cannot be applied to the exact GP model because it will not be an explicit Gaussian likelihood which is non-conjugate and thus we would require approximate inference methods like variational inference or expectation propagation or sample-based schemes like Monte Carlo schemes. From the variational inference approach, there are improved and scalable variational training procedures [Salimbeni et al., 2018, de G. Matthews et al., 2018b, Gardner et al., 2018] which help make this more viable. Other interpretations like

separate transformed functions for the noise model and the GP model are also [Snelson et al., 2003, Maroñas et al., 2020] good options and have seen success in the field. An added bonus is that this method can reap all of the benefits of the inference methods listed above.

Latent Variables

Another approach to account for noise into the inputs is to assume that the inputs are latent variables. We presume to observe the noisy versions $\mathbf{x}$ of the real variable $\bar{\mathbf{x}}$ with some additive noise $\epsilon_{\mathbf{x}}$. We would specify a prior distribution over the function f **and** the inputs $\mathbf{x}$. Using the standard variational approximation procedure [Tran et al., 2016], one could approximate our variational function $\mathbf{f}$ with a variational parameter $q(\mathbf{f})$ as well as our input data $\mathbf{x}$ by some variational parameter $q(\mathbf{x})$. Thus, one could minimize the standard Evidence Lower Bound loss function

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{x}_i)} \left[\mathbb{E}_{q(\mathbf{f}(\boldsymbol{x}_i))} \left[\log p(y_i | \mathbf{f}(\mathbf{x}_i)) \right] \right] - \sum_{i=1}^{N} \mathrm{D}_{\mathrm{KL}} \left[q(\mathbf{x}_i) || p(\mathbf{x}_i) \right]. \tag{3.19}$$

This requires us to calculate an expectation over our likelihood with respect to to our variational distributions $q(\mathbf{f})$ and $q(\mathbf{x})$. This is advantageous because it not only allows one to incorporate training procedures but one could also experiment with different distributions for the input distribution $p(\mathbf{x})$ as well as variational distribution $q(\mathbf{x})$. We typically choose a Gaussian distribution (like in this thesis and the common mean field approximation [Wang and Blei, 2013, Blei et al., 2016]) but there is no restriction on the form of the distribution. From a practical perspective, there are many options for the practitioner to configure the trade-off between the prior configuration and the variational configuration. For example, we could be very loose with our assumptions by initializing the prior with the mean of the noisy inputs and then let both the prior and the variational distribution be a free parameters. Or we could be very strict with our assumptions and set the prior and variational distributions to be our input noise [Damianou et al., 2014, 2016]. From a more bold perspective, one could use normalizing flows [Kobyzev et al., 2019, Papamakarios et al., 2019b] which can offer even richer prior distributions and could help with error propagation even in out-of-sample scenarios [Nalisnick et al., 2019].

As shown in [Villacampa-Calvo et al., 2020], it is possible to utilize the latent variable representation even in classification settings. However, there is no (as far as I know) paper rigorously assessing the impact of constraining this prior distribution, $p(\mathbf{x})$ For example, the paper of [Damianou et al., 2014] features the term "uncertain inputs" but it was never shown how this can be modified in the cases where we **know** the uncertainty of our inputs. There are many combinations that we can use but the interplay between the prior distribution $p(\mathbf{x})$ and the variational distribution $q(\mathbf{x})$ might be difficult to train. This is a potential avenue of exploration especially within the Earth science community.

Deep Gaussian Processes

The original deep GP (DGP) is just a stack of Bayesian GP Latent Variable Models [Damianou et al., 2014] and more recent GPs have incorporated stacking Stochastic Variational GPs [Salimbeni and Deisenroth, 2017], parametric GPs [Jankowiak et al., 2020], Fourier feature representations [Cutajar et al., 2017] or pure Bayesian inference [Havasi et al., 2018]. One research line would be to improve the Bayesian GPLVM in such a way that one can stack the Bayesian GPLVM to constrain the solutions with our known prior distributions. Alternatively, because there is already a lot of stochasticity within the model due to the composition of layers, one could try and apply the augmented predictive mean and variance methodologies as shown in this paper.

3.4. Concluding Remarks

> "*we need to stop jacking around with GPs and actually* ***apply them*** *(Gaussian process methods).*"
> – Neil Lawrence, MLSS 2018

Given the advent of more scalable, *exact* GP methods [Krauth et al., 2017, Raissi, 2017, Wang et al., 2019a], these modifications are immediately available for any standard GP regression method so long as the kernel function is differentiable. So the question is why is this not done in practice regularly? As we have alluded to before, software plays a very important role in how popular ML methods are because it helps shape the way we think and implement solutions to given real problems. By incorporating more functionality and examples in popular GP software packages [Matthews et al., 2017, Gardner et al., 2018], this will popularize the idea of incorporating input uncertainty into GP problems more. Our hope is that didactic packages like *GPJax* or *stheno* will help facilitate future researchers in this aspect. It is very clear from the literature that there are many different approaches to accounting for uncertainty. But in almost every study above, there are not many applications outside of dynamical systems especially within the last decade. It would be great to see applications outside (or within) the community that use and critique these algorithms on different problems. There are many little improvements that can be done within all of the standard GP methods. A simple example that we demonstrated in this thesis is the linearized GP predictive variance estimate that get statistically better variance estimates. This simple augmentation was not very difficult to apply in practice and we hope that our method and similar augmentations will be more widely used in the future.

4. Gaussianization - Information Quantification

Contents

Information theoretic measures are a powerful framework for data analysis as they are non-linear, multivariate, and capture higher-order feature relations. However, one factor that has prevented their wider use is the limiting factor of accurately estimating multivariate densities. In the presence of complex, high-dimensional datasets, many traditional density estimators fail to model the density accurately due to the curse of dimensionality. In this chapter, we motivate the Gaussianization framework amongst the state-of-the-art literature as an excellent candidate density estimator especially in the context of information theory metrics. The included journal article is very complete from theory to practice of how to apply Gaussianization methods to real-world Earth science data. This chapter is concluded with an overview of the strengths and limitations of the paper as well as some future directions.

4.1. Density Estimation

Information theoretic measures provide a good framework for quantifying uncertainty using a myriad of different techniques. Univariate measures such as classic Shannon's entropy and multivariate measures such as joint entropy, mutual information, divergence, and total correlation all provide the user with some metric of comparison across datasets. However, all of these measures are data dependent and require a good estimation of the underlying density $p(\mathbf{x})$. Therefore, having good density estimators is a vital component when considering applying information theory to large scale data. While there have been great papers [Timme and Lapish, 2018] and essays [Kumar and Gupta, 2020] when motivating the use of information theory as an excellent candidate for theory-driven data analysis, addressing the limitations of density estimators for large scale, high-dimensional data has not been the focal point of many explorations. [Timme and Lapish, 2018] successfully made the argument for discrete estimators using robust binning schemes. However, in the context of high-dimensional Earth science data, histograms will fail due to the curse of dimensionality. Other more adaptive binning methods such as kernel density estimation and k-NN density estimators suffer from the curse of dimensionality as well. Therefore, in order to effectively use information theory measures in large scale and real-world datasets, we need to resolve this problem of density estimation which subsequently can help us incorporate information theory into our data analysis schemes.

4.1.1. Generative Modeling

Fortunately, during the last few years the machine learning community has started to put more focus on *generative modeling* in order to produce data distributions instead of just point estimates. Especially within the deep learning community, generative models have become more and more popular as a means to estimate log probabilities and for density sampling. The most popular methods include Variational Autoencoders (VAEs) [Kingma and Welling, 2014], Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] and Normalizing Flows (NFs) [Rezende and Mohamed, 2015]. Each method tries to address the density estimation and sampling problem from a different perspective, depending upon the target application. VAEs are typically used for estimating densities by using an encoder-decoder architecture with fully parameterized neural networks. VAEs and its variants have seen great success in approximate inference schemes. GANs are the method of choice for sampling as it uses a purely generative model via a decoder structure and a discriminator. The loss function incorporates a duality whereby we try to produce good enough samples such that the discriminator cannot tell the difference between the generated samples and real samples. Normalizing Flow models are models use a composition of invertible diffeomorphism functions. One can train this using the exact log-likelihood and subsequently use the *change of variables formula* to sample from your input distribution and also calculate log probabilities. Again, depending upon the target application, one could choose. For more detailed information about these three models please see the following survey papers: VAEs [Kingma and Welling, 2019], GANs - [Creswell et al.,

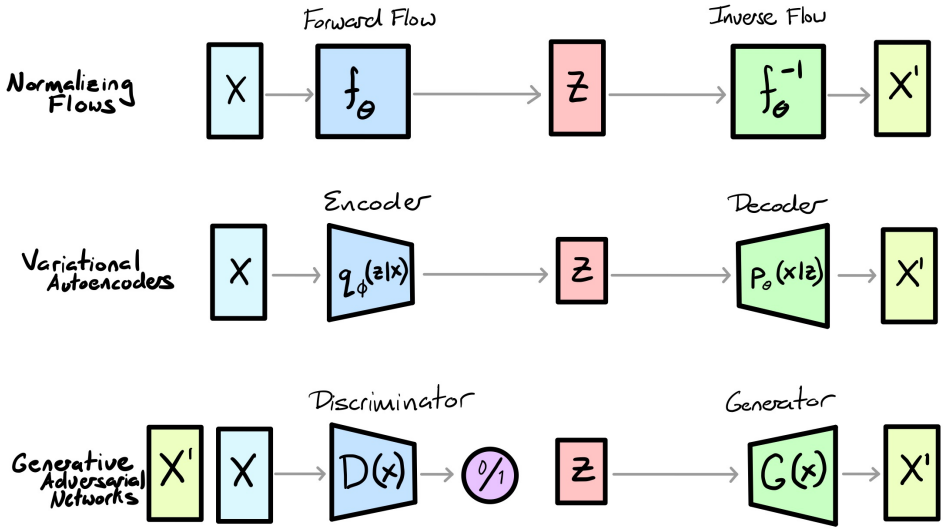


Figure 4.1.: A comparison of the structure for three different popular frameworks. Figure adapted from: lilianweng.github.io.

2018, Wang et al., 2017], NFs - [Kobyzev et al., 2020, Papamakarios et al., 2019a].

All three approaches have shown great promise in the field of generative modeling for machine learning. But, depending upon the target application, one should choose one of the above methods to take advantage of the specific strengths. In the case of density estimation which is necessary for Information theory measures, we need a method that allows for an exact evaluation of probability densities on a new set of points. VAEs use approximate inference and GANs are a pure generative framework which does not do density evaluation at all. Normalizing flows are a framework which allow both exact density estimation and sampling hence we believe it is the best method in this situation. As of now, the NFs community has made a lot of progress with constructing different methods but there has been little attention paid to a particular class of methods that have ties to information theory measures: Gaussianization [Chen and Gopinath, 2001, Laparra et al., 2011a, Meng et al., 2020b]. In the paper, we highlight and motivate the use of Gaussianization as the method of choice not only because it is a good density estimator, but because of the ties to information theory directly through its construction. This framework was originally motivated by information theory and has existed since 2000 which precedes when NFs became popular as they were reintroduced in the neural networks community [Rezende and Mohamed, 2015]. A subsequent paper [Laparra et al., 2011a], in 2011, generalized this framework even further and it is only now, in 2020, that we see the most generalized approach to date [Meng et al., 2020b]. However, all of these papers are focused on the quality of Gaussianization as a density estimator and never in the context of information theory measure estimation. Below, we give a more formal

construction of the parallels between the normalizing flow literature and Gaussianization. The following section will motivate Gaussianization as the idea information theoretic measure.

4.2. Normalizing Flows

Let $\mathbf{Z} \in \mathbb{R}^D$ be a random variable with a tractable PDF: $p_{\mathbf{z}} : \mathbb{R}^D \to \mathbb{R}^D$. The objective is to find some invertible function $\mathbf{f}_{\boldsymbol{\theta}}$ such that $\mathbf{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{z}$. This function is parameterized by $\boldsymbol{\theta}$ which allows us to learn the transformation. Using the change-of-variables formula, we can compute the density of $\mathbf{x}$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = p(\mathbf{f}_{\theta}(\mathbf{x})) \left|\boldsymbol{\nabla}_{\mathbf{x}} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})\right| \tag{4.1}$$

where $\boldsymbol{\nabla}_{\mathbf{x}}$ is the Jacobian of $\mathbf{f}$ and $|\cdot|$ is the absolute determinant. Intuitively, this $|\cdot|$ represents the change in volume of the transformation. f is the normalizing direction as it goes from a more complicated data distribution to a simpler base distribution $p_{\mathbf{z}}$. The inverse function of $\mathbf{f}$, i.e. $\mathbf{f}^{-1}$, is the generative direction as it allows us to sample from $\mathbf{z}$ which we can propagate the samples from the latent space through the function $\mathbf{f}^{-1}$ to get samples in $\mathbf{x}$.

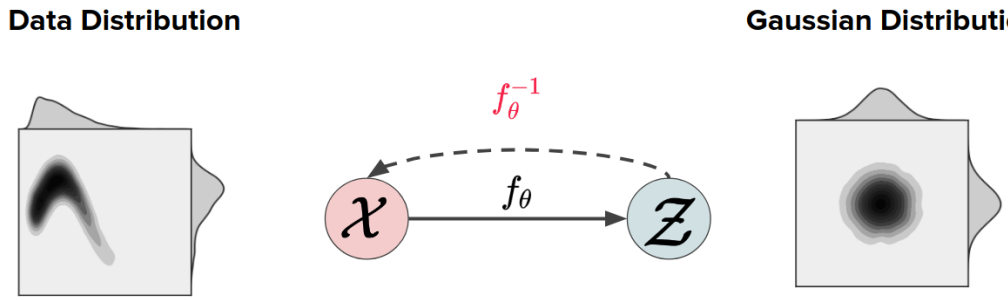


Figure 4.2.: A demonstration of an invertible transformation. We go from complex distribution $\mathcal{X}$ to a latent space $\mathcal{Z}$ which is marginally and jointly Gaussian in this case. The transformation involves an invertible function f, parameterized by θ.

Now the challenge is to design a function $\mathbf{f}(\cdot)$ such that we can learn the mapping from our data to the latent domain. In the case of high-dimensional data, it is nearly impossible to define a transformation expressive enough such that one transformation is enough. Analogous to standard neural networks, we can stack together multiple compositions of simpler arbitrary functions to create more expressive transformations, e.g. figure 4.3. As $\mathbf{f}$ is invertible, we can have $\mathbf{f} = \mathbf{f}_L \circ \ldots \circ \mathbf{f}_1$ which would result in a more expressive transformation. Likewise the inverse is possible $\mathbf{f}^{-1} = \mathbf{f}_1^{-1} \circ \ldots \circ \mathbf{f}_L^{-1}$. The determinant of

the Jacobian in this case is simply the product of all of the transformations $\mathbf{f}_\ell$,

$$|\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x})| = \prod_{\ell=1}^{L} |\nabla_{\mathbf{x}}\mathbf{f}_\ell(\mathbf{x})|\,, \tag{4.2}$$

which gives us more expressivity to transform arbitrarily complex distributions to simpler distributions. Given the correctly chosen set of $\mathbf{f}(\cdot)$, this would be sufficient to estimate any arbitrary distribution [Bogachev et al., 2005, Jaini et al., 2020, Meng et al., 2020a].

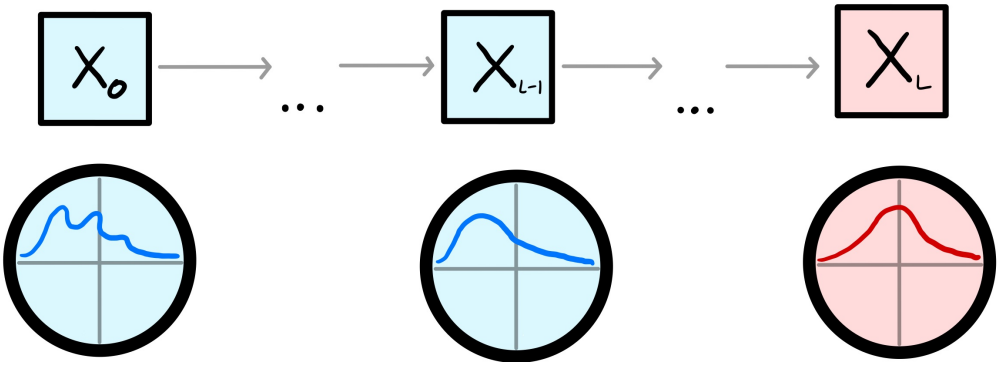


Figure 4.3.: Demonstrates how a composition of invertible functions lead to more expressive transformations. This allows us to use simple transformations that are cheaper to compute. We show the transformation for a 1D case (the histograms). Figure adapted from: lilianweng.github.io.

Because we can evaluate the likelihood exactly, this implies that we can use the negative log-likelihood as a loss criteria:

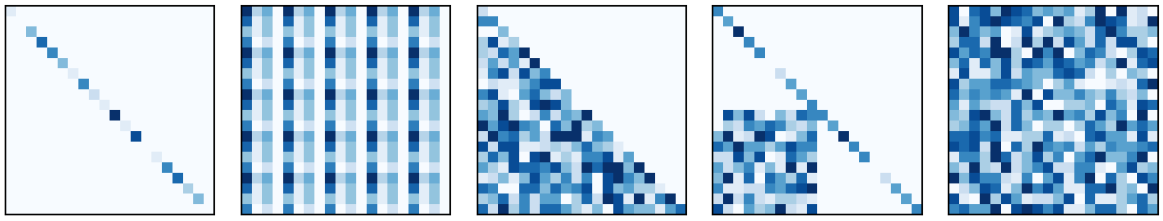
$$\mathcal{L}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}}\left[-\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] = \mathbb{E}_{\mathbf{x}}\left[-\log p(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) - \log|\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})|\right]. \tag{4.3}$$

The standard procedure is to estimate this using Monte Carlo sampling using stochastic gradients. Given the need to constantly evaluate the Jacobian during training, this becomes a bottleneck of this procedure. We want a transformation $\mathbf{f}()$ that is universal i.e. it can approximate any density function arbitrarily well. Hence, the community has put a lot of effort into constructing Jacobian matrices that are easy and cheap to compute yet still expressive enough to learn the complex distribution.

4.2.1. Jacobian Form

As alluded to in the previous section, the bottleneck during the training procedure is evaluating the determinant of the Jacobian.

$$\log p(\mathbf{x}) = \log p\left(\mathbf{f}_\theta(\mathbf{x})\right) + \log \underbrace{|\nabla_{\mathbf{x}}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})|}_{\text{Bottleneck}}. \tag{4.4}$$



(a) Diagonal (b) Low-Rank (c) Triangular (d) Structured (e) Free-Form

Figure 4.4.: A breakdown of the different normalizing flow methods based on the structure of the Jacobian. If we consider the structure of the Jacobian matrix as a transformation's ability to capture the feature-wise dependencies of a dataset, then we see how can go from a least expressive, cheap Jacobian (a) to the most expressive, expensive Jacobian (e). So in principle, one would need more repeated applications of the least expressive methods (a)-(d) compared to the free-form Jacobian (e).

This is an area of intensive research and a really effective way to break down each of the methods [Kobyzev et al., 2020, Papamakarios et al., 2019b]. There are many different (and complicated) frameworks but almost all of them can be put into different categories for how the Jacobian is constructed. A naive full Jacobian matrix is of order $\mathcal{O}(D^3)$. This is fine with simple datasets but this can be prohibitive with datasets with more dimensions. A diagonal Jacobian is $\mathcal{O}(D)$ to evaluate but it lacks the expressivity due to its lack of cross-dimensional considerations. There are also hybrid methods in between these extremes. Figure 4.4 gives a visual reference for the differences. Below we list and briefly highlight each of the Jacobians found in the literature. For a more in-depth breakdown, please see the NF survey literature [Papamakarios et al., 2019b, Kobyzev et al., 2019].

Diagonal

This is known in the NF community as *element-wise* transformations. A function $f(x)$ is applied to each of the features of the dataset. This is analogous to the non-linear layer for neural networks. These Jacobian matrices incorporate the least structure as every transformation has no mixing of variables thus it cannot model correlations between dimensions (figure 4.4 (a)). While it is the least expressive transformation, it is the cheapest and simplest to compute as well, mainly because the determinant of a diagonal matrix is the sum of its diagonal entries. Originally, the NF community used the invertible Leaky ReLU [He et al., 2015] in order to incorporate this non-linearity into Flow models. More recently, there has been a lot of success in Mixture of Gaussian CDF transformations [Ho et al., 2019] as well as spline transformations [Durkan et al., 2019, Dolatabadi et al., 2020]. Both of these methods have shown SOTA results. The Gaussianization Flow [Meng et al.,

2020b] utilizes the Mixture of Logistics (similar to the Mixture of Gaussians) as one of the layers and it has shown competitive results. Because these transformations cannot model correlations between dimensions, they are often coupled with other transformations which do have cross-correlation considerations.

Low Rank

These are Jacobian matrices whose determinant can be easily computed due to some transformation or property which often result in low-rank matrices (figure 4.4 (b)). Some simple examples include orthogonal transformations, e.g. PLU flows [Oliva et al., 2018], QR flows [Hoogeboom et al., 2019], Expontential & Cayley map, and Householder transformations [Tomczak and Welling, 2016]. The Jacobian of these transformations typically have a determinant of exactly ± 1. However, these are the least expressive transformations even when multiple transformations are composed. Thus they are not typically used alone and instead are often coupled with other transformations [Hoogeboom et al., 2019, van den Berg et al., 2018]. Some non-linear transformations, like planar flows [Rezende and Mohamed, 2015] and radial flows [Tabak and Turner, 2013], utilize the matrix determinant lemma [Rezende and Mohamed, 2015] which allows for more efficient computation of the determinant of the Jacobian; often $\mathcal{O}(D)$ instead of $\mathcal{O}(D^3)$. Sylvester flows [van den Berg et al., 2018] extend planar flows to allow for more expressivity by doing an additional matrix multiplications parameterized by an orthogonal transformation. One disadvantage of these low-rank transformations is that they do not have analytical inverse transformations. So many times, these non-linear affine flows are used for variational inference [Rezende and Mohamed, 2015] and not for standard generative models.

Lower Triangular

Another very popular class of models which feature more general neural network architectures are autoregressive functions (AFs) which are constructed by factorizing over the dimension. This results in a lower triangular structure (figure 4.4 (c)) which is cheap determinant calculation $\mathcal{O}(D)$. Some noteable examples include the Invertible AF (IAF) [Kingma et al., 2016], the Neural AF (NAF) [Huang et al., 2018], the Masked AF (MAF) [Papamakarios et al., 2017], and the Block NAF (BNAF) [Cao et al., 2019]. These methods are very flexible and allow the user to use arbitrary neural network architectures within the algorithm which help with expressivity. Both the forward direction $\mathbf{f}_\theta$ and the inverse direction $\mathbf{f}_\theta$ are theoretically equivalent given some conditions [Papamakarios et al., 2017], but one has to be conscious about the application because AFs are dimension sensitive. For example, for density estimation, the standard AF methods are applicable [Papamakarios et al., 2017, Huang et al., 2018, Cao et al., 2019] whereas for sampling, one should use the inverse variant [Kingma et al., 2016].

Structured

These are by far the most popular forms of normalizing flows because they are fairly flexible yet inexpensive to compute. They work by partitioning the transformations such that they are only applied on a subset of dimensions. This results in a structured triangular Jacobian with a block sparse-like structure (figure 4.4 (d)). Because of the structure, the determinant of the Jacobian is as cost efficient as the diagonal Jacobian, $\mathcal{O}(D)$. Some notable examples include the NICE algorithm [Dinh et al., 2015] and its successor Real-NVP [Dinh et al., 2016]. Like AFs, one can use any parameterized NN architecture for the block-regions and these transformations do allow for more feature dependencies to be capture across dimensions yet they do not increase the computational complexity. It also includes one of the most popular methods for image GLOW [Kingma and Dhariwal, 2018], which features 1x1 Convolutional blocks.

Free-Form

The final class of methods features free-form transformations. There is no restriction and thereby is the most expressive transformation in the literature (figure 4.4 (e)). Residual Flows (RFs) [Behrmann et al., 2019, Chen et al., 2019] are based on residual neural network architectures. These have a very expensive Jacobian to calculate but they use a biased stochastic estimate [Behrmann et al., 2019] and an unbiased stochastic estimate via a power series approximation [Chen et al., 2019]. There are also a class of continuous-time flows [Grathwohl et al., 2019] which are based on the neural ODEs literature [Chen et al., 2018]. All free-form methods tend to be more expensive (even with the estimation tricks) and a lot more complicated to implement. But of course the trade-off is that you'll have more expressive Jacobians, and thus will need a lot less layers to effectively learn the probability density function of a difficult dataset.

4.2.2. How do they compare?

As seen in the above section, there is an abundance of methods available for the NF literature. It is, however, very difficult to compare each of them because of the many combinations one could choose. In general, the best method it depends on the task at hand. For tabular datasets (e.g. POWER, GAS, HEPMASS, etc) in the latest review survey [Kobyzev et al., 2019], the Neural Autoregressive Flow (NAF) algorithm [Huang et al., 2018] and the AF with a Spline coupling layer does the best. However, the recent Gaussianization Flows paper [Meng et al., 2020b] (which was not included in the survey) shows substantial improvement for their method over some of the tabular datasets. For the standard image datasets, the Flow++ model [Ho et al., 2019] (which is a make up of convolutions and CDF Mixture Layers) performs substantially better than the other methods. Surprisingly, the free-form methods are not the best despite the fact that they are the most expressive and flexible. Instead, it appears that a combination of simpler

transformations and element-wise transformations like splines or cumulative distribution function (CDF) mixtures seem to perform the best for the standard datasets.

4.3. Different Perspective

The following section is based partially on the work in [Meng et al., 2020b] where they describe the loss function similarities between normalizing flows and Gaussianization, the work in [Papamakarios et al., 2019b] which demonstrates the equivalency between KL-Divergence terms and the work of [Johnson et al., 2020b, Laparra et al., 2020] which highlights the connection between the loss function and Total correlation.

Typically in the normalizing Flows literature, the training loss function, $\mathcal{L}(\boldsymbol{\theta})$, is shown as maximizing the likelihood which we know is equivalent to minimizing the KL-Divergence

$$\underset{\boldsymbol{\theta}}{\text{argmax}}\ \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_{\mathbf{x}}(\mathbf{x};\boldsymbol{\theta})\right] = \underset{\boldsymbol{\theta}}{\text{argmin}}\ \text{D}_{\text{KL}}\left[p_{\text{data}}(\mathbf{x})||p_{\mathbf{x}}(\mathbf{x};\boldsymbol{\theta})\right] + \text{constant} \tag{4.5}$$

However, instead of minimizing the KL-Divergence between the data distribution and our model, we can think of maximizing how well our function f transforms our data $\mathbf{x}$ into a base distribution. It has been shown in in [Papamakarios et al., 2019b, Laparra et al., 2011a], this is equivalent to:

$$\text{D}_{\text{KL}}\left[p_{\text{data}}(\mathbf{x})||p_{\mathbf{x}}(\mathbf{x};\boldsymbol{\theta})\right] = \text{D}_{\text{KL}}\left[p_{\mathbf{z}}\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})\right)||p_{\mathbf{z}}(\mathbf{z}),\right] \tag{4.6}$$

where $p_{\text{target}}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}))$ is the distribution induced by our function $\mathbf{f}_{\boldsymbol{\theta}}$ and $p_{\mathbf{z}}(\mathbf{z})$ is our base distribution. For training this allows us to minimize the KL-Divergence without needing to evaluate the generative direction $\mathbf{f}_{\boldsymbol{\theta}}^{-1} = \mathbf{g}_{\boldsymbol{\theta}}$ which can be costly in some cases [Huang et al., 2018]. So we can still evaluate densities but not necessary generate samples. This is very common in the variational framework [Rezende and Mohamed, 2015] and hence we see a lot of non-invertible flow-models improve the posterior representation [Rezende and Mohamed, 2015, Kingma and Welling, 2019].

This loss term has connections to information theory which allows an entire suite of information-theoretic measures to be used with these models easily. If one explicitly uses a Gaussian distribution as the base distribution, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, then this KL-Divergence term is

$$\text{D}_{\text{KL}}\left[p_{\mathbf{z}}\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})\right)||\mathcal{N}(\mathbf{0}, \mathbf{1})\right] = J(\mathbf{x}), \tag{4.7}$$

which is known as non-Gaussianity; a measure of how far away your distribution is from a Gaussian distribution [Chen and Gopinath, 2001] showed that this divergence term has a different interpretation from an information theory perspective. Using the Pythagorean theorem outlined in [Cardoso, 2003], this can be rewritten as

$$J(\mathbf{x}) = \underbrace{\text{D}_{\text{KL}}\left[p(\mathbf{x})||\prod_{d=1}^{D} p(\mathbf{x}^d)\right]}_{T(\mathbf{x})} + \underbrace{\sum_{d=1}^{D} \text{D}_{\text{KL}}\left[p(\mathbf{x}^d)||\mathcal{N}(0,1)\right]}_{J_m(\mathbf{x})}. \tag{4.8}$$

where D is the dimension of the data. We denote $T(\mathbf{x})$ the total correlation (a.k.a. multi-information, multivariate information) of a multi-dimensional dataset, which measures the dependencies between the features of $\mathbf{x}$. This is a generalization of mutual information. $J_m(\mathbf{x})$ is sum of all marginal non-Gaussianity measures for each components within $\mathbf{x}$. So to minimize this term, we need to minimize both $T(\mathbf{x})$ and $J_m(\mathbf{x})$. To minimize $T(\mathbf{x})$, we would been to make each component statistically independent. This is the objective of Independent Components Analysis (ICA) which maximizes a transformation s.t. the components of $\mathbf{x}$ are independent. This was the basis of the Gaussianization scheme in [Chen and Gopinath, 2001] where they use ICA. In the following paper, we showcase how under a special set of circumstances, we can utilize this connection between the two KL-Divergence term to promote an iterative approach as a generative model with additional benefits involving information theoretic measures.

4.4. The Paper

Title
Gaussianizing the Earth: Multidimensional Information Measures for Earth Data Analysis [Johnson et al., 2020c]

Authors
J. Emmanuel Johnson, Valero Laparra, Maria Piles, Gustau Camps-Valls

4.4.1. Summary

In this paper, we address the solution proposed in section 1.4.3. We propose information theory as a good way to explore high-dimensional datasets as an alternative to tradition methods that only look for linear correlation. We give motivating examples as to why this is with toy example demonstrating that entropy and mutual information are more informative measures with non-linear datasets. We propose Gaussianization as the algorithm of choice to estimate probability density functions as it can deal with the curse of dimensionality. We explain why it is superior to traditional measures for density estimation. We also describe how Gaussianization is especially adept to estimate information theory measures in particular due to the formulation. Gaussianization is simple in principle but can be difficult to understand so care has been taken to really explain the inner mechanics of the algorithm. We give examples of how it can be used in Earth observation applications. We motivate its effectiveness at Gaussianizing data and generate samples for radar and hyperspectral images. We also show the usefulness of total correlation estimates for high resolution aerial imagery to discern their spatial redundancy. We demonstrate how it can be used to assess the relevance and share information between drought variables.

We conclude with a demonstration of how we can assess the affect of spatial-temporal relationships by using information on global datasets.

4.4.2. Contributions

This paper had two main contributions within the scientific community: 1) a viable density estimator for high dimensional problems, and 2) a viable information theoretic estimator. They are not mutually exclusive but it is worth highlighting them separately.

Density Estimation

We presented Gaussianization is a viable density estimator to the Earth science community which deal with high-dimensional, complex datasets. We demonstrated that the Iterative Gaussianization (IG) formulation is simple and easily extendable. The flexibility of IG is that it relies on 1D marginal Gaussianization estimators and orthogonal matrices and we highlighted this with many references to other plug-in-play methods like kernel density estimators (KDE) and principal components analysis (PCA). With toy examples and selected applications within remote sensing to motivate the applied scientists that we have a density estimator that is able to learn probability densities as well as generate samples that are better represented the original data. This paper really does open up the possibilities for Earth scientists because we show that multivariate density estimation is now viable with high-dimensional data. Density estimation is a difficult problem but it opens up many opportunities for future work.

Information Theory Estimator

Information theory measure estimators suffer from practical use in high-dimensional data because they are at the mercy of PDF estimators. We provide a solution that actually leverages these standard estimators in a way that is scalable. In [Laparra et al., 2020], we clearly showed the effectiveness of Iterative Gaussianization over the other classical methods ITEs with toy datasets. Even when estimating information-theoretic measures for a multivariate Gaussian distribution with an exponential family fails with high-dimensions for metrics such as Mutual information and KL-Divergence. For entropy and total correlation, this paper exposes a lot of pitfalls and limitations with standard density estimators. With mutual information and the KL-Divergence, we show that standard methods fail for real distributions at all, with errors over 200 percent even for lower dimensions. In both papers [Laparra et al., 2020, Johnson et al., 2020b], we showcase real world examples ranging from neuroscience to Earth sciences. We demonstrate some targeted applications such as drought monitoring indices and how one can go about selecting the most informative variables and spatial-temporal representations based on entropy and mutual information. We also do a grand-scale application where we assess the spatial-temporal entropy of variables from the Earth system data cube [Mahecha et al., 2020] across the globe over

the span of years 2008-2010. This kind of analysis is not possible with standard methods yet we were able to find interesting patterns across spatial-temporal resolutions and variables.

4.4.3. Reproducibility

Code Base

All of the available code for the paper is available at the following url[1]. This includes all of the toy experiments and the real experiments that used the ESDC dataset [Mahecha et al., 2020]. This is the main repo for the majority of the paper which was done in Python. It also features a webpage[2] with full-fledged demos. In addition, there is a preliminary implementation for fully parameterized Gaussianization flows is available[3]. Effort was made to integrate this method into mature normalizing flow packages like *nflows* [Durkan et al., 2020]. It is a slightly different implementation than the one found in [Meng et al., 2020b] but it is a much cleaner implementation that uses all of the components found within the *nflows* package. We hope this will provide a starting point for improving upon and incorporating more advanced components.

Datasets

For the datasets used in the applied experiments, we use the Earth System Data Cubes [Mahecha et al., 2020] which are available from the Earth System Data Lab[4]. After signing up for access, it is free to use.

Software

There was already a MATLAB version for the Gaussianization algorithm[5]. This was the basis for the new python package[6] that was developed during this thesis. It follows the standard *scikit-learn* [Varoquaux et al., 2015] API which is easy to use and familiar to the machine learning community. We also attempted to merge our methods to a more general package for *Density-Destructive-Learning* [Inouye and Ravikumar, 2018] which features a more broad class of iterative Normalizing flow methods. So a fork of the library was created with demos and convenient wrappers for the RBIG method in particular[7]. To try and fix some of the speed issues we encountered during the large scale usage, a

[1] https://github.com/IPL-UV/gauss4eo
[2] https://ipl-uv.github.io/gauss4eo/
[3] https://github.com/IPL-UV/gaussflow
[4] https://www.earthsystemdatalab.net/
[5] https://github.com/IPL-UV/rbig_matlab
[6] https://github.com/IPL-UV/rbig_jax
[7] https://github.com/jejjohnson/destructive-deep-learning

preliminary implementation JAX [Bradbury et al., 2018] back-end is also available[8] which will allow one to use GPUs and TPUs.

Literature

Gaussianization is not often referenced within the Normalizing flows community surveys [Papamakarios et al., 2019b, Kobyzev et al., 2019] as a primary algorithm within the family of methods. To contribute into this line of research and development, and to also promote the use of Gaussianization methods in broader communities, we become a primary contributor to a continuously updated normalizing flow literature repository[9].

4.5. Further Research Directions

During the publication process of the journal article listed in the previous section, further research was done and as mentioned in the reproducibility section, most of these ideas are documented on the literature review page[10]. In remainder of the chapter, we will summarize some of the highlights of the literature and give suggestions for further research.

4.5.1. Limitation of Iterative Approach

The original message in [Laparra et al., 2011a] was that density estimation and information theory metrics suffer from the expensive evaluation of the Jacobian. Since then, the normalizing flow community have demonstrated methods to mitigate this and have subsequently had a lot of success for many different disciplines [Kobyzev et al., 2019, Papamakarios et al., 2019b]. However, the case could be made that iterative methods [Laparra et al., 2011a, Inouye and Ravikumar, 2018] could be a more viable approach for estimating information theory metrics. Gaussianization was constructed on principle using the KL-Divergence in the transform domain. Subsequently, they found that a composition of element-wise and orthogonal transformations resulted in guarantees of convergence which could be shown iteratively. So naturally, an iterative technique with a stopping criteria makes sense in the case. However, having a fully parameterized solution Meng et al. [2020a] is the most generalized method of Gaussianization to date which took inspiration from the normalizing flows literature by taking the parameterized mixture of Gaussians and the Householder flows [Tomczak and Welling, 2016]. Subsequently, they found that they were able to get much higher quality density estimators than the iterative approach [Laparra et al., 2011a]. So if we have empirical evidence that shows that the iterative approach is inferior for the quality in the transform domain, we can pose the

[8] https://github.com/IPL-UV/rbig_jax
[9] https://github.com/janosh/awesome-normalizing-flows
[10] https://jejjohnson.github.io/uncertain_gps/Notes/literature/

question of how the iterative approach is still useful. Below we highlight some strengths and weaknesses between both methods.

Strengths

Simplicity. This method is relatively simple to implement. We cannot underestimate the simplicity making it very attractive to use for experts and non-experts alike. It is iterative, so users will not have to deal with too many issues related to neural networks and gradients such as initialization, learning rate, vanishing/exploding gradients. The new worry is more about convergence and how does one effectively determine the best stopping criteria.
Backwards Compatibility. You can use any plug-in-play estimator of your choice available from a wide range of methods in the literature. For the side of the marginal density estimator, we have methods that are piece-wise such as histogram, smooth like kernel density estimation and adaptive like k-Nearest Neighbours. At the end of the day, there are many very good methods for estimating 1D densities. One thing to note is that all of the problems are inherited while doing this such as boundaries and parameters. For the side of the random orthogonal rotation, we have methods like ICA [Chen and Gopinath, 2001] and, thanks to [Laparra et al., 2011b], we can use any orthogonal estimator like PCA or random rotations.
Guaranteed Convergence. There are some theoretical guarantees that given enough successive transformations, the resulting distribution will be Gaussian; [Chen and Gopinath, 2001] for ICA and [Laparra et al., 2011b] for any random rotation. This means that you are guaranteed to converge to a Gaussian distribution as long as you use a marginal Gaussianization and orthogonal rotation. This was also extended to the fully parameterized version [Meng et al., 2020a]. Not all methods in the normalizing flow literature can offer such strong guarantees in theory **and** in practice.

Weaknesses

Inefficient. This is the biggest weakness of the iterative approaches: they are not efficient. Inevitably one needs a lot of layers even for simple transformations because it is such a basic method. It means that it will require a lot of memory when there are a lot of layers especially with very high dimensional data. A key factor is there is no batch processing since these methods do not have any sort of stochastic approximations or gradient-based training. So with datasets as large as 1 million points with 1000 dimensions, iterative techniques will suffer as we have personally seen during the course of this thesis. This also prevents these methods from using modern hardware such as Graphical Processing Units (GPUs) do to memory constraints.
Same Problems as 1D Density Estimators. Just like there are issues with 1D density estimators, the same problems exist on a large scale in Gaussianization schemes. Boundaries at 1D PDF functions which affect outliers are still relevant, except there are many many more of them. Each 1D PDF estimator will have parameters and these are still present in

the Gaussianization methods as well. And since they are not optimized for the dataset, they can affect many parts of your algorithm, e.g. the quality of the density estimation, the quality of your generated samples, and the values of the information theory metrics. This algorithm can be sensitive to the decision you do make.
Error Accumulation. This is very related to the above issues but it becomes very apparent with iterative techniques: any errors or problems accumulate with each iteration. As there is no global optimization strategy to correct any weights or parameters, the errors can accumulate quickly. In addition, if you use a stopping criteria that does not have anything to do with the optimization, then that can lead to even more issues; again with specific parts of the method e.g. the sampling, the density or the metrics.

4.6. Concluding Remarks

Having a viable estimator of multivariate densities, and hence of information theory measures like mutual information, opens up many opportunities for more exploration and hypothesis testing. A very simple extension is to extend the literature for measuring the similarity between distributions. There are many similarity metrics that do not directly estimate the density function $p(\mathbf{x})$. This include known linear methods such as the RV coefficient [Robert and Escoufier, 1976], distance measures such as the distance correlation [Székely et al., 2007], and kernel measures such as Hilbert-Schmidt Independence Criterion [Gretton et al., 2005]. They all wish to evaluate some similarity between two datasets. Mutual information is not normally included within the standard methods is notoriously difficult to estimation. From a model comparison standpoint, we now have a method which can serve as a benchmark when evaluating other parameterized metrics on well-known distributions such as the multivariate Gaussian. For example, we can apply this to compare CMIP model simulations [Correa and Lindstrom, 2013b, Eyring et al., 2016, Bruinsma et al., 2020] which can be extended to other model comparison scenarios where we go beyond standard correlations. We could also use it to analyze neural network architectures to find similarities [Kornblith et al., 2019, Tang et al., 2020] and the information bottleneck [Ma et al., 2020, Ardizzone et al., 2020].

In terms of algorithmic improvements, the dimensionality is still an issue as this can occur within ultra-dimensional datasets like images and video. For the Gaussianization scheme, this will require convolutions Hoogeboom et al. [2019] which has yet to be explored in this work due to the orthogonality constraints from the Iterative Gaussianization framework [Laparra et al., 2011a]. An alternative approach is to use surjective flows [Nielsen et al., 2020] which allow one to reduce the dimensionality with a simple change to the Jacobian. This has been applied in other applications involving non-Euclidean domains whereby the authors showcase density estimation on other surfaces including a globe[Rezende et al., 2020]. We also look forward to seeing comparisons the ITMs produced by Gaussianization to other NF architectures as well as completely different approaches using neural networks [Belghazi et al., 2018, Chan et al., 2019]. In general, we are certain that this contribution will help the community apply ITMs in practice now

that we have better density estimators within the machine learning literature that can deal with large-scale, high-dimensional data.

5. Discussion and Conclusions

Machine learning has made great strides in today's science and engineering. However, quantifying the uncertainty and information content within our system's data and models are still unresolved problems. These issues hamper the wide use and adoption of current models, especially with the advent of more heterogeneous and multivariate data sources. In addition, many of these datasets exhibit high levels and variety of noise and uncertainties. In this thesis we introduce different methodologies to deal with the problems of uncertainty and information quantification in arbitrary multivariate datasets.

The definition of information has a foundation in uncertainty. By quantifying the uncertainty, we are actually quantifying our knowledge or ignorance about the data generating process, which in turn translates into information. Uncertainty can come in many forms: 1) data, 2) model, and 3) out-of-sample. In terms of uncertainty quantification, Bayesian methods tend to capture the inverse direction; related to the uncertainty in the model parameters. However, many machine learning models do not adequately capture the forward direction, i.e. error propagation, as they either omit the input uncertainty altogether or there is simply an inadequate treatment. Once we have an adequate density representation for our data, we can use information theory measures to summarize the information content. One can quantify not only the uncertainty of the data and model, but also summarize and compare different representations via characterizing the shared information across different datasets.

This process is also increasingly complex when we consider spatial-temporal-spectral datasets which are multivariate and high-dimensional. This type of data is very prevalent in Earth science applications which poses many challenges for machine learning methods. There are many decisions to be made when we consider which is the best model capture relationships between variables. Consider an ML pipeline where we want to model some relationships between two datasets. On the front end, we could use more spatial features than temporal features. Or we could also perform dimensionality reduction to get the best lower dimensional representation. But how do we assess whether the chosen representation is the most *informative*? Further down the ML pipeline, we also want to use eventually noisy, biased, multivariate datasets as inputs to models, and we need to ensure that their uncertainty is correctly propagated through the models. But how do we propagate these uncertainties? and furthermore how do we assess our predictive variance? And then finally, is there some post-analysis we can do to assess the models to ensure they are properly validated and calibrated? Are there further changes one can make in either the input space or model definition? These are all decisions that we make when we apply machine learning methods in practice irregardless of how we modify the framework using physics-informed decisions.

5.1. Themes and Contributions

Generally ML problems fall under three learning paradigms: 1) supervised discriminate machine learning finds a single best set of parameters to describe the approximate conditional distribution, 2) probabilistic supervised machine learning finds a distribution of parameters that best describe the conditional distribution, and 3) probabilistic unsupervised machine learning seeks to find a joint distribution described by an underlying latent variable. All three formulations are valid and we have seen incredible results for many Earth science applications. There are many things to consider when choosing an approach irrespective of the application; things like computational budget, data complexity, and even expertise. Not everyone can be Bayesian and not every application calls for estimating joint densities. In this thesis, we looked at all three different approaches to machine learning and improved some of the underlying problems that we often find within practical applications.

5.1.1. Part 1: Sensitivity Analysis in Kernel Methods

The most popular form of machine learning is to use discriminative models. This entails finding an approximate conditional distribution $p(\mathbf{y}|\mathbf{x})$ parameterized by a function, f_θ. If we try to learn a single set of parameters for this function, we are not accurately characterizing the model uncertainty nor the data uncertainty. While this method does not take into account uncertainty, there are methods that attempt to augment the training procedure, methods that train multiple models, and post-hoc methods that allow one to analyze the variance and the relative relevance of the input features. We chose sensitivity analysis as it closely rivals the literature for physical models. In particular, we chose to focus on derivative-based measures as they are fairly simple and closely follow the original definition of sensitivity analysis. However, there is insufficient literature and exploration about the parallels and differences between how one can apply sensitivity analysis for physical models, and how one can apply sensitivity analysis for machine learning models in general. For example, every parameter within a physical model is well-motivated and has a clear physical meaning. In non-parametric ML models, however, when trying to find functions' parameters, they are often inaccessible. So we sought to explore a family of non-parametric methods called kernel methods where we would dig deeper into the formulation and intuition to see how derivative-based analysis can help us explain the inferred models.

Contribution

We chose kernel methods because they are very flexible models which provide non-linear extensions to many of the classic machine learning models. Consequently, that allowed us to explore not only kernel regression and kernel classification problems, but also kernel density estimation and kernel independence measures. In addition, kernel methods have many connections to other methods such as neural networks and probabilistic models

such as Gaussian processes so our work can be inspiration for derivative-based sensitivity analysis for other related ML algorithms. Kernel methods are often described as "black-box" models because we do not have access to the feature map explicitly. However, we showed that derivative-based analysis can help open that box and give insight into the models' decisions.

In the paper [Johnson et al., 2020b], we focused on intuition and we demonstrated all claims with many toy examples for each of the disciplines. Because of the formulation, we were able to draw parallels between all of the core kernel methods. For each of the kernel methods (regression, classification, density estimation and independence), we gave some motivating examples of how the derivatives can help with understanding. For example, in regression and classification we showed how one can use sensitivity analysis to get feature relevance and sample attribution. In kernel density estimation, we showed that the derivatives can help visualize the density ridge. And in kernel independence measures, we showed how derivatives help visualize the change of direction within the dependencies between two variables. We also showed proof-of-concept examples using spatial-temporal EO data with toy examples and real examples. For example, we showed how a regression problem could be formulated to find the sample attribution for different configurations of spatial-temporal Earth data inputs, and we showed how the solution to a binary drought detection problem using Support Vector Machines can be analyzed to reveal the sample attribution which reveals the boundary between classes.

Our experiments were not perfect. For example, in chapter 2, we highlighted the fact that we made general assumptions about smoothness by only using the RBF kernel function and did not experiment with other more expressive kernels which is a big limitation of kernel methods in common practice [Krauth et al., 2017]. We also did not look at the sensitivity of the hyperparameters for the unsupervised kernel methods. Unsupervised methods are notorious for using ad-hoc procedures to find the kernel parameters which could have adverse affects on the results we obtained. In addition, real problems involving spatial-temporal datasets are often at a much larger scale so the effectiveness of our analysis may not translate to problems of that scale. However, despite all this, we firmly believe that the work will be well received by the community for any applications involving kernel derivatives. The advent of automatic differentiation eases the burden of calculating derivatives so we foresee many applications using the derivative of scalable kernel methods in the future.

5.1.2. Part 2: Error propagation in Gaussian Processes

Bayesian methods allow users to fully describe the model with probabilistic measures. In supervised settings, users define a prior distribution over their parameters and a likelihood describing the generating process for their inputs to their outputs. Then, by normalizing by the evidence (i.e. the data), they get a posterior which describes the best set of parameters given the observed data. These methods feature predictive uncertainty which is a combination of the aleatoric and epistemic uncertainty. This enables one to get not

only mean predictions but also confidence intervals. In this thesis, we focused on Gaussian processes, which are typically the golden standard for confidence intervals [Wilson and Izmailov, 2020]. However, while these methods inherently handle the uncertainty in the (hyper-)parameters, they do not typically account for the uncertainty within the input data. For example, as mentioned in chapter 3, the predictive variance in the standard GP formulation does not depend on the inputs which is not ideal for noisy inputs. There are cases where we want to propagate the noisy inputs through our learned function. There are many approaches to doing so but they are typically not used in practice, especially within the Earth science community.

Contribution

In chapter 3, we went through the formulation of the GP and showed how to modify the posterior predictive mean and variance of a standard Gaussian process to account of input uncertainty. We took inspiration from dynamic GPs [Girard et al., 2002b, Deisenroth, 2010, McHutchon and Rasmussen, 2011], which iteratively update their predictions with every time step. We showcased how a similar formulation can be used in non-dynamical settings. We assumed a GP has already been trained and consider the case where we have noisy inputs and we want to propagate this through the posterior, i.e. the predictive mean and variance. We showcase two distinct Gaussian approximation methods to achieve this: 1) a linearized version via a Taylor's expansions and 2) a moment-matching approximation method. The Taylor expansion is faster as it only depends on the derivative whereas the moment-matching approximation requires expectations of kernel functions. Both methods produce better predictive variance estimates than the standard GP for toy datasets.

In the paper [Johnson et al., 2020a], we showed how the linearized approximation can be applied in practice. The example featured noisy ultra-spectral reflectance input values from the Infrared atmospheric sounding interferometer (IASI) [Chalon et al., 2001] satellite and we fit an Gaussian process to regress the global surface temperature values. One reason we focused on the linearized approximation is because moment-matching is expensive with higher dimensions [Deisenroth and Mohamed, 2012]. The other reason is because the linearized approximation is a much simpler formulation as it involves only the derivative of the predictive mean function. Despite the fact that it is a simpler method, we showed that the method performed better than the standard GP both qualitatively and quantitatively. We found that there was a stronger relationship between the global temperature maps for the predictive variance and the absolute error for our predictions and this was confirmed via a qualitative inspection.

The field is wide open for future studies regarding input uncertainty for Bayesian models. Based on this work, there should be more studies showcasing more applications where these methods succeed or even fail. We also welcome more studies into the computational aspects which would be useful for surrogate models to construct better emulators [Rivera et al., 2015, Svendsen et al., 2020]. For example, the one computational

burden for the moment-matching method is to approximate the kernel expectations via deterministic sigma-point estimation [de Souza et al., 2020]. Furthermore, one can easily apply the same formulation to sparse Gaussian processes [Bui et al., 2017b], which will enable these methods to scale to much larger datasets. Another front is to consider noisy inputs when training. The most promising method features the Bayesian Gaussian Process Latent Variable Model [Damianou et al., 2014, Villacampa-Calvo et al., 2020] which assumes the inputs as latent variables defined as a distribution. This allows one to train a GP model with the noisy inputs which will have an effect on the learned model parameters. Another very promising method is the emergence of Deep GPs [Salimbeni and Deisenroth, 2017, Cutajar et al., 2017]. These models are a composition of GPs where the outputs from one GP is the input to the next which is inherently a stochastic process. This is an interesting avenue which has ties to other methods like conditional density estimation and covariate models [Dutordoir et al., 2018]. While our first paper on this work was rather simplistic, we hope that its simplicity will attract more users to apply GPs on even more complex tasks with noisy datasets.

5.1.3. Part 3: Gaussianization for Information Theory Metrics

Estimating an arbitrary multivariate density of your data is a harder problem than the previous approaches. It assumes no function to describe the relation between the marginals and instead assumes that one can describe the entire joint distribution using some parameterized function of an underlying latent distribution. If one can successfully describe the joint density, then it may also be possible to describe the conditionals as well as the marginals which are essentially a generalization of the previous approaches. Another advantage of this approach is that once we have a probabilistic density function of our data, we can use information theory measures to summarize and make comparisons across models. Once we have a model for the density of our data, IT gives us access to different approaches to quantify the information content by summarizing the uncertainty using different measures. Measures like entropy and total correlation can be used to calculate the average uncertainty as well as the redundancy. Other measures like mutual information and KL-divergence allows us to make (relative) comparisons across different datasets. While information theory measures are popular as components within machine learning algorithms, they are generally not widely adopted in many applications.

The main issue is the problem of density estimation for high-dimensional datasets. Most classical methods like probabilistic PCA or ICA are not expressive enough to model complex datasets and many non-parametric estimators like histograms and kernel density estimators fail due to the curse of dimensionality. Recently there has been a family of methods called normalizing flows [Kobyzev et al., 2019, Papamakarios et al., 2019b] methods which have had great success in many ML applications including images, audio and video. They utilize the change of variables formulation which allows one to not only evaluate the density of your data but also generate samples. While this potentially solves the high-dimensional density estimation problem, this thesis dives deeper in a particular

formulation of normalizing flows which has direct connections to information theory.

Contribution

In chapter 4, we motivated the use of normalizing flows as compared to the classical methods (e.g. non-parametric density estimation) and other standard generative models such as Variational AutoEncoders and Generative Adversarial Networks. We went through the formulation for NFs and highlighted the fact that most methods in the literature can be categorized based on the properties of the Jacobian. The determinant of the Jacobian is the bottleneck of the NF method because it is naively of $\mathcal{O}(D^3)$ with D dimensions. So many of the methods feature a trade-off between a more expressive and expensive Jacobian versus a less expressive and cheaper Jacobian. We also showcased how the log-likelihood loss function can related to the KL-Divergence in the original domain $\mathcal{X}$ and the transform domain $\mathcal{Z}$ which adds some flexibility as to how these are trained. We also showed how this simple relationship allows one to express the loss function in terms of Information theory metrics, i.e. the total correlation and the marginal non-Gaussianity measure. We hope that this explicit connection will be the inspiration for other practitioners to investigate other clever schemes or decompositions that might reveal insights into the relationship between Normalizing flows and Information theory measures.

In the paper [Johnson et al., 2020b], we motivated the use of Gaussianization as a simple iterative case of Normalizing flows. We decomposed the algorithm to really highlight each of its components (an orthogonal rotation followed by an element-wise, marginal Gaussianization) to motivate how this method can be easily extended to feature more powerful transformations that are present in the NFs literature. Lastly, we showcased how iterative Gaussianization is uniquely qualified to produce information theory metrics due to its formulation [Laparra et al., 2020]. For intuition, we motivated IT metrics over classical correlation metrics with toy examples showcasing their superiority for comparing non-linear and complex datasets. In addition, we wanted to stress that having access to the marginal entropy, joint entropy, and the mutual information paints a more complete picture of the entire relationship with regard to the complexity and noise rather than just one of those measures.

For applications, we demonstrated its viability as a density estimator by producing samples that were marginally and jointly representative for a Hyperspectral image. We also gave an example of how entropy and mutual information can be used to determine the best temporal feature representation for different drought indicators. This experiment is very representative of the power of information theory as a measure to help decide on the most appropriate feature representation which can be used as inputs for other ML models. Lastly, we performed a more ambitious experiment to show how we could compare different spatial-temporal feature representations of different Earth system variables. We show different ways one could display the results for further insight like global information maps which can be further summarized by entropy. We were successful in il-

lustrating that different spatial-temporal feature representations exhibit different patterns both in the global maps as well as the entropy curves.

While we were successful in our experiments, we also highlighted some of the pitfalls and future work that needs to be done to make this a more competitive and well-adopted method within the community. The iterative approach is very simple and keeps all assumptions fixed with every iteration until convergence. On the practical side, this is disadvantageous because it results in excess layers which produces numerical errors. On the theoretical side, this may be an incorrect assumption because the 1D Gaussianization models used at the top layer are probably not the same assumptions we would use in later layers. Overall, this can lower the quality of the sampling and density estimation. The fully parameterized Gaussianization scheme [Meng et al., 2020b] is a reflection of this analysis because they produced significantly lower log-likelihood estimates with fewer layers in their experiments. However, we showed that the iterative Gaussianization scheme produced the best information theoretic measures for controlled datasets compared to all of the canonical methods like k-NN. This makes the Gaussianization scheme a unique and competitive contender for estimating information theoretic measures. While its current formulation is relatively slow for high-dimensional datasets, we are confident that we can make speedups to make it a more viable candidate in daily use.

In the normalizing flows literature, we have seen the parameterized Gaussianization method [Meng et al., 2020b] compare favourably to the more popular NF methods like Real-NVP and Autoregressive-Spline models. The next step would be to see how well the IT metrics using the iterative methods compare to the fully parameterized solutions. For really high-dimensional datasets (i.e. more than 500 dimensions), we also need to include convolutional layers to really capture the dependencies between the marginal distributions for datasets such as images.

5.2. Future Work

In each of the above sections, we highlighted the contributions in terms of their individual approaches with respect to uncertainty quantification and information content. However, one aspect we wish to promote within this thesis is the idea that the methods presented are not mutually exclusive. There is a lot of overlap between the methods as one could possibly incorporate one set of techniques into other frameworks to overcome some of the inherent limitations of said framework. Below we highlight some promising examples based on the findings of this thesis that would be appropriate for future work.

Sensitivity Analysis and Error Propagation. The SA methods presented were motivated by the fact that it is a viable option in the discriminative framework due to the limitations that discriminative models exhibit for uncertainty quantification. However, we also highlighted within chapter 2 and in [Johnson et al., 2020b] that one can use the same derivative-based analysis in other probabilistic kernel methods like Gaussian process regression or even GP latent variable models [Titsias and Lawrence, 2010]. While sensitivity

analysis has been used for GPs in previous works [Blix et al., 2017] for identifying feature relevance, there has not been any exploration for sensitivity analysis for GPs that propagate input error. There also have not been many other papers utilizing SA for other more advanced GP algorithms. In addition, one could also use SA in other previously suggested GP methods which incorporate the input errors during the training phase like stochastic variational GPs [Bui et al., 2017b], Bayesian GPLVMs [Titsias and Lawrence, 2010, Damianou et al., 2016, Villacampa-Calvo et al., 2020] and Deep GPs [Salimbeni and Deisenroth, 2017, Cutajar et al., 2017, Havasi et al., 2018]. This is an interesting avenue because one is already characterizing the input distribution with prior knowledge and thus SA can be applied in a more principled way. One could also use other SA approaches outside of derivative-based measures that are variance-based [Sobolá, 2001] or even model agnostic methods [Da Veiga, 2015, Lundberg and Lee, 2017b]. This approach would have a complete UQ quantification in the forward and inverse direction for estimating conditional distributions using machine learning.

Gaussianization and Sensitivity Analysis. In sensitivity analysis, we need to find an appropriate distribution for the inputs to propagate through the learned model to measure how much it affects the output variance. In many approaches like the traditional Sobol indices methods, we assume that the features are independent [Sobolá, 2001] and generate samples based on their marginal distributions. While this approach is much simpler, it is often not correct as many high-dimensional, multivariate datasets have many dependencies between the features that are not taken into account for the models. We have shown that Gaussianization (and other NF methods) is an effective density estimator for multivariate datasets which do produce samples that are marginally and jointly consistent with the original dataset. These methods could produce multivariate representative samples that could be used with any SA method. One could do experiments with different input data representations (or jointly with the outputs) to see how the output variance is affected. This would give practitioners better options than the standard Monte Carlo schemes which have been shown to be limiting in high-dimensional settings [Razavi et al., 2020].

Gaussianization and Latent Variable Models. As mentioned in chapter 3, we assumed that the GP model is already trained and a method to overcome this limitation is to use GP Latent Variable Models [Titsias and Lawrence, 2010]. One foreseeable limitation is the expressivity of the variational distribution used to approximate the prior we impose for our input data and our posterior. Normalizing flows have already shown great promise in similar applications to help improve the expressivity of the posterior approximation VAEs [Rezende and Mohamed, 2015] and also in Monte Carlo sample-based inference [Hoffman et al., 2019, Wu et al., 2020]. In this application, one could augment the variational distribution to be more expressive [Maroñas et al., 2020] or we could transform (or warp) our inputs/outputs using NFs as density estimators during the training phase [Lalchand et al., 2021]. So while we are still using GPs, we are effectively using

density estimators on either end of our datasets which is an interesting approach that may allow an even better data representation and uncertainty characterization.

5.3. Parting Thoughts

In this thesis, we demonstrated several machine learning approaches (kernel methods, Gaussian processes and multivariate Gaussianization) to handle uncertainty and information quantification. Methods were extensively validated in a wide diversity of Earth system science problems, involving many types of learning problems (classification, regression, density estimation, synthesis, error propagation and information-theoretic measures estimation), sensory data (radar, multispectral, hyperspectral, infrared sounders), data products (observations, reanalysis and model simulations) and resolutions (in space, time and spectrum). From a more theoretical standpoint, we also showed how there are many connections and none of the methods are mutually exclusive. We hope that this motivates more researchers to use our methods and investigate even better alternatives methods that can be used for even more difficult challenges than what we presented. We also hope that there will be more publications on data uncertainty and information content in newer additions within the scientific machine learning literature [Willard et al., 2020, Jia et al., 2020].

ML research can be very daunting these days and it can take effort to really bridge the gap across communities. Applied machine learning is very difficult and we need more scrutiny when applying these methods to real data. We cannot just apply methods without thinking about the implications. The skepticism from domain experts is warranted which motivated us to really explain our steps in a way that is accessible to domain experts at all levels. These are exciting times to be in applied ML and these past few years, we have seen many potential of ML in applied settings [Reichstein et al., 2019, Watt-Meyer et al., 2021, Payrovnaziri et al., 2020, Arrieta et al., 2020, Brajard et al., 2020, Willard et al., 2020, Jia et al., 2020] as well as the pathologies that we need to overcome [Yang and Perdikaris, 2019, Dennis et al., 2019, Wilson and Izmailov, 2020]. We look forward to future endeavours and the progress we will see in the applied settings stemming from the methods discussed within this thesis.

Acknowledgements

The research activities leading to this thesis were supported by the European Research Council (ERC) through the ERC-CoG-2014 SEDAL Project under Grant 647423 and the ERC-SyG-2019 USMILE project (grant agreement 855187). I thank the European Space Agency (ESA) for support via the Early Adopter Call of the Earth System Data Lab project in 2018-2019. I also thank R. Jenssen for providing guidance and accommodation during my distance stay at UiT, Tromso, Norway in 2020.

5.4. Published Work

Journal Articles

1. **J. E. Johnson**, V. Laparra and G. Camps-Valls, "Accounting for Input Noise in Gaussian Process Parameter Retrieval", in *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 3, pp. 391-395, March 2020, doi: 10.1109/LGRS.2019.2921476.

2. **J. E. Johnson**, V. Laparra, A. Pérez-Suay, M. D. Mahecha, G. Camps-Valls, "Kernel methods and their derivatives: Concept and perspectives for the earth system sciences", in *PLOS ONE* 15(10): e0235885, October 2020, doi: 10.1371/journal.pone.0235885.

3. **J. E. Johnson**, V. Laparra, Maria Piles, and G. Camps-Valls, "Gaussianizing the Earth: Multidimensional Information Measures for Earth Data Analysis", in *IEEE Geoscience and Remote Sensing Magazine*, 2021, doi: 10.1109/MGRS.2021.3066260.

Related Journal Articles

- R, Sauzède, **J. E. Johnson**, H. Claustre, G. Camps-Valls, A. B. Ruescas, "Estimation of Oceanic Particulate Organic Carbon with Machine Learning", *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*; Gottingen Vol. V-2-2020, (2020): 949-956. DOI:10.5194/isprs-annals-V-2-2020-949-2020

- V. Laparra, **J. E. Johnson**, R. Santos-Rodriguez, G. Camps-Valls, J. Malo, "Information Theory Measures via Multidimensional Gaussianization", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (**Submitted**)

- K. Blix, A. Ruescas, **J. E. Johnson**, G. Camps-Valls, "Learning Relevant Features of Optical Water Types", in *IEEE Geoscience and Remote Sensing Letters*, 2021, doi: 10.1109/LGRS.2021.3072049.

Conference Papers

- **J. E. Johnson**, V. Laparra and G. Camps-Valls, "Disentangling Derivatives, Uncertainty and Error in Gaussian Process Models," IGARSS 2018 - *2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia, 2018, pp. 4051-4054, doi: 10.1109/IGARSS.2018.8519020.

- **J. E. Johnson**, V. Laparra, R. Santos-Rodriguez, G. Camps-Valls, J. Malo, " Information Theory in Density Destructors", in International Conference of Machine Learning Workshop, June 2019.

- **J. E. Johnson**, S. Sundaresan, T. Daylan, L. Gavilan, D. K. Giles, S. Ishitani Silva, A. Jungbluth, B. Morris, "RotNet: Fast and Scalable Estimation of Stellar Rotation

Periods Using Convolutional Neural Networks", *NeurIPS 2020*, Machine Learning for Physical Sciences Workshop,

5.5. A Note on Reproducibility

Reproducible Code

Chapter 2 (Experimental Repo) https://github.com/IPL-UV/sakame

Chapter 2 (Software) https://github.com/IPL-UV/jaxkern

Chapter 3 (Experimental Repo) https://github.com/IPL-UV/gp_error_propagation

Chapter 3 (Software) https://github.com/jejjohnson/uncertain_gps/

Chapter 3 (Literature Review) https://github.com/jejjohnson/uncertain_gps/

Chapter 3 (Literature Review) https://github.com/jejjohnson/gp_model_zoo

Chapter 4 (Experimental Repo) https://github.com/IPL-UV/gauss4eo

Chapter 4 (Software) https://github.com/IPL-UV/rbig

6. Summary in Valencian

6.1. Motivació

L'aprenentatge automàtic ha fet grans avenços en la ciència i l'enginyeria actuals. No obstant això, quantificar la incertesa i el contingut de la informació dins de les dades i models del nostre sistema encara són problemes sense resoldre. Aquests problemes dificulten un ús més ampli i l'adopció de models actuals, especialment amb l'aparició de fonts de dades més heterogènies, provinents de molts sensors i multivariants. A més, molts d'aquests conjunts de dades presenten alts nivells i varietat de soroll i incerteses. En aquesta tesi doctoral introduïm diferentes metodologies per tractar els problemes d'incertesa i quantificació de la informació en conjunts de dades multivariants arbitraris.

La definició d'informació té un fonament formal en el concepte d'incertesa. Al quantificar la incertesa, en realitat estem quantificant el nostre coneixement o desconeixement sobre el procés de generació de dades, que al cap i a la fi es tradueix en informació. La incertesa pot presentar-se de moltes formes: 1) incertesa a les dades, 2) incertesa al model i 3) incertesa fora de la mostra. En termes de quantificació d'incertesa, els mètodes bayesians tendeixen a captar la quantificació d'incertesa inversa, què està relacionat amb la incertesa en els paràmetres del model. No obstant això, molts models d'aprenentatge automàtic no capturen adequadament la incertesa directa, ja que ometen la incertesa d'entrada o simplement hi ha un tractament estadístic inadequat. Una vegada tenim una representació de densitat de probabilitat adequada per a les nostres dades, la teoria de la informació ens proporciona una metodologia directa per resumir el contingut d'informació d'un conjunt de dades arbitrari. Es pot quantificar no només la incertesa de les dades i el model, sinó també resumir i comparar diferents representacions mitjançant la caracterització de la informació compartida entre diferents conjunts de dades.

Aquest procés també és cada vegada més complex quan considerem conjunts de dades espacial-temporals-espectrals que són multivariants i d'alta dimensió. Aquest tipus de dades és molt freqüent en aplicacions de ciències de la Terra, la qual cosa suposa molts desafiaments per als mètodes d'aprenentatge automàtic actuals. Hi ha moltes decisions a prendre quan considerem el desenvolupament d'algorismes, i una d'elles -tal volta la més crucial- és establir quines variables son les més rellevants i quines son les relacions entre les observacions disponibles. Per exemple, a l'hora de desenvolupar una cadena de processament basada en ML volem dissenyar-la de forma robusta, de forma que no hi haga redundància entre variables, que s'hi identifiquen clarament les components espacials i temporals, i que el contingut expressiu (informatiu) siga el màxim per a la tasca final que afrontem. Tal volta al principi seleccionem característiques, però d'altres vegades estem

interessats en fer una reducció de dimensionalitat per obtenir una millor (i més compacta) representació de característiques de baixa dimensionalitat. Però, com avaluem si la representació escollida és la més *informativa*? Encara més sovint l'objectiu implica utilitzar conjunts de dades multivariants, eventualment sorolloses i esbiaixadess als models, i hem d'assegurar-nos que la seva incertesa es propaga correctament als models. Però, com propagem aquestes incerteses? I, a més a més, com avaluem la variància predictiva de les nostre prediccions? I, finalment, hi ha cap postanàlisi que puguem fer per avaluar els models i així garantir que estiguen correctament validats i calibrats? Hi ha més canvis que es poden fer a l'espai d'entrada o a la definició del model per tal d'adreçar-ho? Totes aquestes són les decisions que prenem quan apliquem mètodes d'aprenentatge automàtic a la pràctica, i que tenen implicacions molt rellevants al disseny teòric dels algorismes.

6.1.1. Contribucions

Generalment els problemes de ML es troben sota tres paradigmes d'aprenentatge: 1) l'aprenentatge automàtic supervisat discriminatiu on es tracta de trobar un millor conjunt de paràmetres per descriure la distribució condicional aproximada; 2) l'aprenentatge supervisat probabilístic on es tracta d'aprendre una distribució de paràmetres que millor descriuen la distribució condicional, i 3) l'aprenentatge automàtic probabilístic sense supervisió que busca una distribució conjunta que siga descrita per variables latents subjacents. Les tres formulacions són vàlides i hem vist resultats excel·lents a moltes aplicacions de les ciències de la Terra. Hi ha molts assumptes a tenir en compte a l'hora d'escollir un enfocament o un altre independentment de l'aplicació; coses com ara el cost computacional, la complexitat de les dades i fins i tot l'experiència de l'usuari. No tothom pot ser bayesià i no totes les aplicacions requereixen estimar la densitat de les densitats de probabilitat. En aquesta tesi, hem examinat els tres enfocaments diferents de l'aprenentatge automàtic i hem millorat alguns dels problemes subjacents que sovint trobem dins de les aplicacions pràctiques.

6.2. Part 1: Anàlisi de sensibilitat en mètodes nucli

La forma més popular d'aprenentatge automàtic és utilitzar models discriminatius. Això implica trobar una distribució condicional aproximada $p(\mathbf{y}|\mathbf{x})$ parametritzada per una funció, $f_{\,theta}$. Si intentem aprendre un únic conjunt de paràmetres per a aquesta funció, no estem caracteritzant amb precisió la incertesa del model ni la incertesa de les dades. Tot i que aquest mètode no té en compte la incertesa, hi ha mètodes que intenten augmentar el procediment d'entrenament, mètodes que entrenen múltiples models i mètodes post-hoc que permeten analitzar la variància i la rellevància relativa de les característiques (variables) d'entrada. En la tesi hem decidit centrar-nos a l'anàlisi de sensibilitat ja que rivalitza amb la literatura dels models físics. En particular, vam optar per centrar-nos en mesures basades en derivades, ja que són bastant simples i segueixen de prop la definició original d'anàlisi de sensibilitat. No obstant això, no hi ha prou literatura i exploració so-

bre els paral·lelismes i les diferències entre com es pot aplicar l'anàlisi de sensibilitat per a models físics i com es pot aplicar l'anàlisi de sensibilitat per a models d'aprenentatge automàtic en general. Per exemple, tots els paràmetres d'un model físic estan ben motivats i tenen un significat físic clar. En els models de ML no paramètrics, però, quan s'intenta trobar els paràmetres de les funcions, sovint són inaccessibles i no tenen un significant físic clar. Així doncs, hem explorat una família de mètodes no paramètrics anomenats mètodes nucli ('kernel methods') on aprofundim en la formulació i la intuïció per veure com l'anàlisi basada en derivades ens pot ajudar a explicar els models inferits.

6.2.1. Contribució

Hem escollit mètodes nucli perquè són models molt flexibles que ofereixen extensions no lineals a molts dels models d'aprenentatge automàtic clàssics. En conseqüència, això ens va permetre explorar no només problemes de regressió i classificació, sinó també d'estimació de densitats i mesures d'independència. A més a més, els mètodes nucli tenen moltes connexions amb altres, com ara les xarxes neuronals i models probabilístics com els processos gaussians, de manera que el nostre treball pot ser una inspiració per a l'anàlisi de sensibilitat basat en derivades per a altres algorismes de ML relacionats. Els mètodes nucli sovint es descriuen com a models de "caixa negra" perquè no tenim accés explícit a la funció de mapeig. Tanmateix, vam demostrar que l'anàlisi basada en derivades de la funció de decisió pot ajudar a obrir aquesta caixa i donar informació sobre les decisions dels models.

A l'article [Johnson et al., 2020b], ens vam centrar en la intuïció i vam demostrar totes les afirmacions amb molts exemples joguina per a cadascuna de les disciplines. A causa de la formulació, vam poder establir paral·lelismes entre tots els mètodes bàsics del nucli. Per a cadascun dels mètodes nucli (regressió, classificació, estimació densitats i independència), hem donat alguns exemples motivadors de com les derivades poden ajudar a comprendre allò que han après els models. Per exemple, a la regressió i classificació vam mostrar com es pot utilitzar l'anàlisi de sensibilitat per obtenir rellevància de funcions i atribució de mostres. En l'estimació de la densitat del nucli, vam demostrar que l'anàlisi de sensibilitat dels kernels pot ajudar a visualitzar les corbes principals de les densitats. I a les mesures d'independència entre variables aleatòries, vam mostrar com les derivades ajuden a visualitzar el canvi de direcció dins de les dependències entre dues variables. També vam mostrar exemples reals utilitzant dades d'observació de la terra amb característiques espacial-temporals. Per exemple, vam mostrar com es podia formular un problema de regressió per trobar l'atribució de mostres per a diferents configuracions d'entrades de dades terrestres espacial-temporals, i vam mostrar com es pot analitzar la solució a un problema de detecció de sequera mitjançant màquines de suport vectorial (SVM en anglès), atribuint una rellevància a les mostra, el qual ha revelat una relació directa amb el marge de separació entre classes.

Els nostres experiments no van ser perfectes. Per exemple, al capítol 2, vam destacar el fet que vam fer suposicions massa generals sobre la suavitat de les funcions de forma

implícita al fer servir el nucli RBF, i no vam experimentar amb altres nuclis més expressius [Krauth et al., 2017]. Tampoc no vam examinar la sensibilitat dels hiperparàmetres per als mètodes nucli sense supervisió. Els mètodes sense supervisió són notoris per utilitzar procediments ad-hoc per trobar els paràmetres del nucli que podrien tenir efectes adversos en els resultats obtinguts. A més, els problemes reals que impliquen conjunts de dades espacial-temporals solen ser a una escala molt més gran, de manera que l'eficàcia de la nostra anàlisi potser no es traduesca necessàriament en problemes d'aquesta escala. Malgrat tot, creiem fermament que el treball serà ben rebut per la comunitat per a aplicacions que impliquen interpretabilitat en mètodes kernel. L'arribada de la diferenciació automàtica alleuja la càrrega del càlcul de derivades, de manera que preveiem moltes aplicacions que utilitzen la derivada de mètodes nucli en el futur.

6.3. Part 2: Propagació d'errors en processos gaussians

Els mètodes bayesians permeten els usuaris descriure completament el model amb mesures probabilístiques. En configuracions supervisades, els usuaris defineixen una distribució prèvia sobre els seus paràmetres i una probabilitat que descriu el procés de generació de les seues dades. Després, normalitzant-se mitjançant l'evidència (és a dir, les dades), obtenen una probabilitat a posteriori que descriu el millor conjunt de paràmetres donades les dades observades. Aquests mètodes presenten una incertesa predictiva, que és una combinació de la incertesa aleatoria i la epistèmica. Això permet obtenir no només prediccions mitjanes, sinó també intervals de confiança. En aquesta tesi, ens hem centrat en els processos gaussians, que normalment són els mètodes estàndard per obtenir intervals de confiança en problemes de regressió no paramètrica [Wilson and Izmailov, 2020]. Tanmateix, tot i que aquests mètodes gestionen intrínsecament la incertesa dels (hiper)paràmetres, no solen tenir en compte la incertesa de les dades d'entrada. Per exemple, tal com s'esmenta al capítol **??**, la variància predictiva en la formulació GP estàndard no depèn de les entrades, i això és una deficència greu perquè en la majoria dels problemes les dades d'entrada son sorolloses i subjectes a incertesses. Hi ha casos en què volem propagar les entrades sorolloses a través de la nostra funció de procés gaussià apresa. Hi ha molts enfocaments per fer-ho, però normalment no s'utilitzen a la pràctica, especialment dins de la comunitat de ciències de la Terra.

6.3.1. Contribució

Al capítol 3, vam examinar la formulació del GP i vam mostrar com modificar la mitjana predictiva posterior i la variància d'un procés gaussià estàndard per tenir en compte la incertesa d'entrada. Ens vam inspirar en els GPs dinàmics [Girard et al., 2002b, Deisenroth, 2010, McHutchon and Rasmussen, 2011], que actualitzen iterativament les seves prediccions amb cada pas de temps. Vam mostrar com es pot utilitzar una formulació similar en entorns no dinàmics. Suposem un GP prèviament entrenat i considerem el cas en què tenim entrades sorolloses i volem propagar-les per obtenir la probabilitat a posteriori, és

a dir, la mitjana predictiva i la variància. Al treball mostrem dos mètodes d'aproximació gaussians diferents per aconseguir-ho: 1) una versió linealitzada mitjançant expansions de Taylor i 2) un mètode d'aproximació de "moment matching". L'expansió de Taylor és més ràpida ja que només depèn de la derivada, mentre que l'aproximació de moment requereix calcular esperances de funcions del nucli. Tots dos mètodes produeixen millors estimacions de variància predictiva que el GP estàndard per als conjunts de dades de joguines.

A l'article [Johnson et al., 2020a], vam mostrar com es pot aplicar l'aproximació linealitzada a la pràctica. L'exemple mostrava valors sorollosos d'entrada de la reflectància ultra-espectral provinents del sensor IASI [Chalon et al., 2001], i ajustem un procés gaussià per fer el modelat invers i poder estimar els valors de temperatura superficial globalment. Una de les raons per les quals ens hem centrat en l'aproximació linealitzada és que el mètode d'ajustament de moments és computacionalment costosa amb dimensions més elevades [Deisenroth and Mohamed, 2012]. L'altra raó és perquè l'aproximació linealitzada és una formulació molt més senzilla, ja que només implica la derivada de la mitjana predictiva. Tot i que és un mètode més senzill, vam demostrar que el mètode funciona millor que el GP estàndard tant qualitativa com quantitativa. Vam trobar que hi havia una relació més forta entre els mapes globals de temperatura per a la variància predictiva i l'error absolut de les nostres prediccions, cosa que es va confirmar mitjançant una inspecció qualitativa.

El camp està molt obert per a futurs estudis sobre incertesa d'entrada per als models bayesians. Basant-se en aquest treball, hi hauria d'haver més estudis que mostrin més aplicacions en què aquests mètodes tenen èxit o fins i tot fracassen. També hi manquen més esforços sobre els aspectes computacionals, els quals seran ben útils per a crear millors emuladors [Rivera et al., 2015, Svendsen et al., 2020]. Per exemple, l'única càrrega computacional per al mètode de "moment matching" és aproximar les esperances del nucli mitjançant una estimació determinista del punt sigma [de Souza et al., 2020]. A més a més, es pot aplicar fàcilment la mateixa formulació a processos gaussians dispersos ("sparse" [Bui et al., 2017b], que permetran escalar aquestos mètodes a conjunts de dades molt més grans. Un altre front obert és el de considerar les entrades sorolloses a l'hora d'entrenar els GPs. El mètode més prometedor sembla ser el de variable latent del procés gaussià bayesià [Damianou et al., 2014, Villacampa-Calvo et al., 2020] que assumeix les entrades com a variables latents definides com a distribució. Això permet entrenar un model GP amb les entrades sorolloses, què tindran un efecte sobre els paràmetres del model apresos. Un altre mètode molt prometedor és l'aparició de Deep GPs [Salimbeni and Deisenroth, 2017, Cutajar et al., 2017]. Aquests models són una composició de GPs on les sortides d'un GP són l'entrada al següent, que és inherentment un procés estocàstic, però no necessàriament un procés gaussià. Aquesta és una via interessant que té vincles amb altres mètodes com l'estimació de densitat condicional i els models covariables [Dutordoir et al., 2018]. Tot i que el nostre primer treball sobre aquest treball va ser un poc simplista, esperem que la seva senzillesa atraurà més usuaris a aplicar GPs en tasques encara més complexes amb conjunts de dades sorollosos.

6.4. Part 3: Gaussianització per a l'estimació de mesures de la teoria de la informació

Estimar una densitat multivariada arbitrària a partir de dades observacionals és el problema més important i difícil en estadística i ML, i té implicacions en totes les aproximacions als enfocaments anteriors. En tots els casos, una aproximació no paramètrica no assumeix cap funció per descriure la relació entre les distribucions marginals i, en canvi, assumeix que es pot descriure tota la distribució conjunta utilitzant alguna funció parametritzada d'una distribució latent subjacent. Si es pot descriure amb èxit la densitat de probabilitat, també podria ser possible descriure les densitats condicionals i les marginals, què són essencialment una generalització dels enfocaments anteriors. Un altre avantatge d'aquest enfocament és que un cop tenim una bona estimació de la funció de densitat probabilística de les nostres dades, podem utilitzar mesures de la teoria de la informació per resumir-la i fer comparacions entre models. Un cop tenim un model per a la densitat de les nostres dades, les mètriques de la teoria de la informació (TI) ens donen accés a diferents aproximacions per quantificar el contingut de la informació i resumir la incertesa a les dades. Es poden utilitzar mesures com ara l'entropia i la correlació total per calcular la incertesa mitjana i la redundància. Altres mesures com la informació mútua i la divergència de Kullback-Leibler (KLD) ens permeten fer comparacions (relatives) entre diferents conjunts de dades. Tot i que les mesures de la teoria de la informació són populars com a components dels algorismes d'aprenentatge automàtic, generalment no s'adopten àmpliament en moltes aplicacions.

El problema principal és el problema de l'estimació de densitat per a conjunts de dades d'alta dimensió. La majoria dels mètodes clàssics com el PCA probabilístic o l'ICA no són prou expressius per modelar conjunts de dades complexos i molts estimadors no paramètrics basats en histogrames i els estimadors de densitat fracassen a causa de l'anomenada "maledicció de la dimensionalitat" ("curse of dimensionality" en anglès). Recentment hi ha aparegut una família de mètodes anomenats Normalizing Flows (NFs) [Kobyzev et al., 2019, Papamakarios et al., 2019b] que han tingut un gran èxit en moltes aplicacions de ML, incloses imatges, àudio i vídeo. Utilitzen el canvi de formulació de variables que permet no només avaluar la densitat de les vostres dades, sinó també generar mostres. Tot i que això potencialment resol el problema d'estimació de densitat d'alta dimensió, aquesta tesi s'endinsa en una formulació específica de NFs que té connexions directes amb la teoria de la informació.

6.4.1. Contribució

Al capítol 4, hem motivat l'ús dels NFs en comparació amb els mètodes clàssics (per exemple, estimació de densitat no paramètrica) i altres models generatius estàndard com ara els autoencoders variacionals i les xarxes generatives adversarials. Hem revisat la formulació dels NF i hem destacat el fet que la majoria de mètodes de la literatura es poden classificar en funció de les propietats del jacobià. El determinant del jacobià és

el coll d'ampolla del mètode NF perquè és ingenuament de $\mathcal{O}(D^3)$ amb dimensions de D. Molts dels mètodes presenten una compensació entre un jacobià més expressiu i costós front un jacobià menys expressiu i més eficient computacionalment. També hem demostrat que la funció de pèrdua basada en la inversemblança es pot relacionar amb la divergència de KL al domini original $\mathcal{X}$ i al domini transformat $\mathcal{Z}$, que afegeix certa flexibilitat quant a la forma en què s'entrenen els models. També hem mostrat com aquesta simple relació permet expressar la funció de pèrdua en terme de mesures de la teoria de la informació com ara la correlació total i la mesura marginal de no gaussianitat anomenada negentropy. Esperem que aquesta connexió explícita siga la inspiració per a altres professionals per investigar esquemes o descomposicions alternatives que puguen revelar informació sobre la relació entre els NFs i les mesures de la teoria de la informació.

A l'article [Johnson et al., 2020b], vam motivar l'ús de la gaussianització com a un cas particular iteratiu de NFs. Vam descomposar l'algoritme per ressaltar realment cadascun dels seus components (una rotació - transformació ortogonal - seguida d'una gaussianització marginal) es pot ampliar fàcilment per presentar transformacions més potents que estan presents a la literatura dels NFs. Per últim, vam mostrar com la gaussianització iterativa està qualificada de manera única per estimar mesures de teoria de la informació a causa de la seua formulació tan genèrica i flexible [Laparra et al., 2020]. Per a obtindre certa intuïció de la transformada proposta, vam motivar la estimació de mesures de TI en exemples joguina que mostren la seva superioritat per comparar conjunts de dades no lineals i amb relacions complexes. A més a més, volíem subratllar que el fet d'accedir a l'entropia marginal, a l'entropia conjunta i a la informació mútua dóna una visió més completa de tota les relacions pel que fa a la complexitat i el soroll.

Pel que fa les aplicacions, vam demostrar la seva viabilitat com a estimador de densitats multidimensionals, produint mostres que eren representatives marginalment i conjuntament d'una imatge hiperespectral. També vam donar un exemple de com es pot utilitzar l'entropia i la informació mútua per determinar la millor representació temporal de les característiques usades àmpliament com a indicadors de sequera. Aquest experiment és molt representatiu del poder de la teoria de la informació com a mesura per ajudar a decidir la representació de característiques més adequada que es pot utilitzar com a entrada a models de ML. Finalment, vam realitzar un experiment més ambiciós per mostrar com podríem comparar diferents representacions espacial-temporals de diferents variables del sistema terrestre. De fet vam demostrar diferents maneres en què es podrien mostrar els resultats per obtenir més informació, com ara mapes d'informació global que es poden resumir amb la entropia. Els resultats demostren que diferents representacions de característiques espacial-temporals presenten diferents patrons tant als mapes globals com a les corbes d'entropia, i fan per tant que la caracterització dels senyals siga completa.

Tot i que vam tenir èxit en els nostres experiments, també vam destacar algunes de les limitacions i el treball futur que cal fer perquè aquest sigui un mètode més competitiu i útil per a la comunitat. L'enfocament a la gaussianització iterativa és molt senzill i manté fixats tots els supòsits amb cada iteració fins a la convergència. Pel que fa a la pràctica, això és desavantatjós, ja que resulta en capes excessives que produeixen una acumulació

d'errors numèrics. Pel que fa a la part teòrica, això pot ser un supòsit incorrecte perquè els models de gaussianització unidimensionals utilitzats a la capa superior probablement no són els mateixos supòsits que utilitzaríem en capes posteriors. En general, això pot reduir la qualitat del mostreig i l'estimació de la densitat. L'esquema de gaussianització completament parametritzat [Meng et al., 2020b] és un reflex d'aquesta anàlisi perquè en general produeixen estimacions de probabilitat significativament més baixes amb menys capes. Tanmateix, hem demostrat que l'esquema iteratiu de gaussianització produeix les millors mesures teòriques de la informació per a conjunts de dades sintètiques al ser comparat amb tot un seguit de mètodes canònics d'estima com el k-NN. Això fa que l'esquema de gaussianització proposat en esta tesi puga ser considerat un candidat únic i competitiu per estimar mesures teòriques de informació. Tot i que la seva formulació actual és relativament costosa computacionalment per conjunts de dades d'alta dimensió, estem ben segurs que podem fer acceleracions perquè siga un candidat més viable en l'ús diari.

A la literatura de NFs, hem vist que el mètode de gaussianització parametritzat [Meng et al., 2020b] es compara favorablement amb els mètodes NF més populars com els models Real-NVP i Autoregressive-Spline. El següent pas seria veure la comparació de les mètriques de TI que utilitzen els mètodes iteratius amb les solucions completament parametritzades. Per als conjunts de dades realment d'alta dimensió (és a dir, més de 500 dimensions), també hem d'incloure capes de convolució per captar les dependències entre les distribucions marginals en dominis estructurats com ara imatges o vídeos

6.5. Treball futur

En cadascuna de les seccions anteriors, hem destacat les contribucions dels distints mètodes proposats pel que fa a la quantificació de la incertesa i el contingut de la informació en dades i models. No obstant això, un aspecte que volem promoure en aquesta tesi és la idea que els mètodes presentats no s'exclouen mútuament. Hi ha molta superposició entre ells, ja que es podria incorporar un conjunt de tècniques a altres marcs per superar algunes de les limitacions inherents en qualsevol d'ells. A continuació, destaquem alguns exemples prometedors basats en els resultats d'aquesta tesi que serien adequats per a futurs treballs.

Anàlisi de sensibilitat i propagació d'errors. Els mètodes d'anàlisi de sensibilitat (SA, en anglès) presentats van estar motivats pel fet que és una opció viable en el marc discriminatiu a causa de les limitacions que presenten estos models per quantificar la incertesa. Tanmateix, també hem destacat al capítol **??** i a [Johnson et al., 2020b] que es pot utilitzar la mateixa anàlisi basada en derivades en altres mètodes probabilístics com la regressió de processos gaussians o fins i tot models de variables latents [Titsias and Lawrence, 2010]. Tot i que l'anàlisi de sensibilitat s'ha utilitzat per a GPs prèviament [Blix et al., 2017] per tal d'identificar la rellevància de les variables, no hi ha hagut cap exploració per a l'anàlisi de sensibilitat de GPs que propaguen l'error d'entrada. Tampoc no hi ha hagut massa articles que utilitzen SA per a altres algorismes de GP més avançats. A

més, també es podria utilitzar SA en altres mètodes de GP suggerits prèviament que incorporen els errors d'entrada durant la fase d'entrenament com els GPs variacionals estocàstics [Bui et al., 2017b], GPLVM Bayesian [Titsias and Lawrence, 2010, Damianou et al., 2016, Villacampa-Calvo et al., 2020] i els Deep GPs [Salimbeni and Deisenroth, 2017, Cutajar et al., 2017, Havasi et al., 2018]. Aquesta és una via interessant perquè com que ja s'està caracteritzant la distribució d'entrada amb coneixement previ, el SA es podria aplicar d'una manera més fonamentada. També es podrien utilitzar altres aproximacions de SA en mesures basades en derivades basades sobre la variància predictiva [Sobolá, 2001] o fins i tot modelar mètodes agnòstics [Da Veiga, 2015, Lundberg and Lee, 2017b]. Aquest enfocament tindria una quantificació de la incertesa completa en la direcció directa i la inversa per tal d'estimar distribucions condicionals amb aprenentatge automàtic.

Gaussianització i anàlisi de sensibilitat. En l'anàlisi de sensibilitat hem de trobar una distribució adequada perquè la incertesa de les entrades es propague a través del model per estimar la incertesa en la variància de sortida. En molts enfocaments com els mètodes tradicionals d'índexs de Sobol, suposem que les característiques són independents [Sobolá, 2001] i generem mostres basades en les seves distribucions marginals. Tot i que aquest enfocament és molt més senzill, sovint no és correcte, ja que molts conjunts de dades multivariades d'alta dimensió tenen moltes interdependències. Hem demostrat que la gaussianització (i altres mètodes de NFs) és un estimador de densitat eficaç per a conjunts de dades multivariades que produeixen mostres que són coherents marginalment i conjuntament amb el conjunt de dades original. Aquests mètodes podrien produir mostres representatives multivariades que es podrien utilitzar amb qualsevol mètode SA. Es podrien fer experiments amb diferents representacions de dades d'entrada (o conjuntament amb les sortides) per veure com afecta això a la variància de sortida. Això donaria als professionals més bones opcions que els esquemes estàndard de Monte Carlo, que han demostrat estar limitats en problemes d'alta dimensió [Razavi et al., 2020].

Gaussianització i models de variables latents. Com es va esmentar al capítol **??**, hem assumit en tot moment que el model GP ja està entrenat i que un mètode per superar aquesta limitació és utilitzar models de variables latents de GP [Titsias and Lawrence, 2010]. Una limitació previsible és l'expressivitat de la distribució variacional que s'utilitza per aproximar el prior que imposem per a les nostres dades d'entrada i la nostra distribució a posteriori. Els NFs ja han demostrat ser prometedors en aplicacions similars per ajudar a millorar l'expressivitat de les aproximacions posteriors basades en VAEs [Rezende and Mohamed, 2015] i també a la inferència basada en mostres de Monte Carlo [Hoffman et al., 2019, Wu et al., 2020]. En aquesta aplicació, es podria augmentar la distribució variacional per ser més expressiva [Maroñas et al., 2020] o podríem transformar (o deformar) les nostres entrades / sortides utilitzant NFs com a estimadors de densitat durant la fase d'entrenament [Lalchand et al., 2021]. Per tant, encara que utilitzem GPs, en realitat estariem utilitzant efectivament estimadors de densitat a l'entrada o sortida, la qual cosa és una aproximació que podria permetre una representació de dades i una caracterització d'incertesa encara millors.

6.6. Conclusions

En aquesta tesi hem proposat diversos enfocaments d'aprenentatge automàtic (mètodes nucli, processos gaussians i gaussianització multivariada) per tractar la incertesa i la quantificació de la informació. Els mètodes es van validar en una àmplia diversitat de problemes científics en ciències de la Terra, que impliquen molts tipus de problemes d'aprenentatge (classificació, regressió, estimació de densitats i dependències, síntesi, propagació d'errors i estimació de mesures de la teoria de la informació), gran diversitat de sensors (radar, multiespectral, hiperespectral, sondes d'infrarojos), productes de dades (observacionals, reanàlisi i simulacions de models) amb distintes característiques i resolucions (en espai, temps i espectre).

Des d'un punt de vista més teòric, també hem demostrar com hi ha moltes connexions i que cap dels mètodes s'exclouen mútuament. Esperem que això motivi a més investigadors a utilitzar els nostres mètodes i investigar alternatives encara millors que es puguen utilitzar per a reptes encara més difícils dels que hem presentat ací. També esperem que hi haga més publicacions sobre la incertesa a les dades i el contingut informatiu amb possiblement nous desenvolupaments a la literatura científica d'aprenentatge automàtic [Willard et al., 2020, Jia et al., 2020]. L'aprenentatge automàtic aplicat son molt difícils, i necessitem més deteniment a l'hora d'aplicar aquestos mètodes a dades reals. No podem aplicar mètodes sense pensar en les asuncions i en les implicacions. L'escepticisme dels experts en dominis està garantit, cosa que ens ha motivat a explicar realment els nostres passos d'una manera accessible als experts en dominis a tots els nivells. Són moments emocionants per treballar en el desenvolupament de ML aplicat i, en els darrers anys, hem vist moltes possibilitats en l'ús d'estes eines a les ciències de la Terra i el Clima [Reichstein et al., 2019, Watt-Meyer et al., 2021, Payrovnaziri et al., 2020, Arrieta et al., 2020, Brajard et al., 2020, Willard et al., 2020, Jia et al., 2020], però també obstacles de tota mena que cal superar com la interpretabilitat dels models i l'estima de la informació i incertesa que hem tractat en esta tesi [Yang and Perdikaris, 2019, Dennis et al., 2019, Wilson and Izmailov, 2020]. Esperem esta tesi servisca de punt de partida per a futurs esforços en la monitorització del planeta.

A. Uncertainty Quantification

> No one trusts a model except the man who wrote it; everyone trusts an observation except the man who made it.
>
> Harlow Shapely

A.1. Definition of Uncertainty Quantification

Uncertainty is an unavoidable feature that we will face in science and modeling. The reality is as follows: our *observations* (the input data and the output data) can only be measured within a limited amount of accuracy and our *models* will always be an approximation to reality. In other words, there will always be error in our datasets and models. While the scientific community agrees that uncertainty quantification (UQ) is very important, there are competing definitions in the literature [Razavi and Gupta, 2015a]. One very complete definition is as follows:

> *"UQ (Uncertainty Quantification) studies all sources of error and uncertainty, including the following: systematic and stochastic measurement error; ignorance; limitations of theoretical models; limitations of numerical representations of those models; limitations of the accuracy and reliability of computations, approximations, and algorithms; and human error. A more precise definition is UQ is the end-to-end study of the reliability of scientific inferences."*
> –Scientific Grand Challenges for National Security: The Role of Computing at the Extreme Scale [Sullivan, 2015]

There is a lot to unpack within this definition, but it does capture the notion: to describe and characterize the error we see in our modeling pipeline so that we make better, more informed decisions and adjustments. UQ will not tell us if our model is correct or not, but it can give us a better indication of trustworthiness and validity. From there, the users can make subsequent decisions about what to do with the models' results. This is very apparent in many applications where we have models in critical domains like the medical field, autonomous driving and also Earth science.

Before we highlight the different types of uncertainty quantification, one first needs to state the sources of uncertainty in order to make decisions about which approach to take. Below, we describe the sources of uncertainty within a modeling perspective.

A.2. Sources of Uncertainty

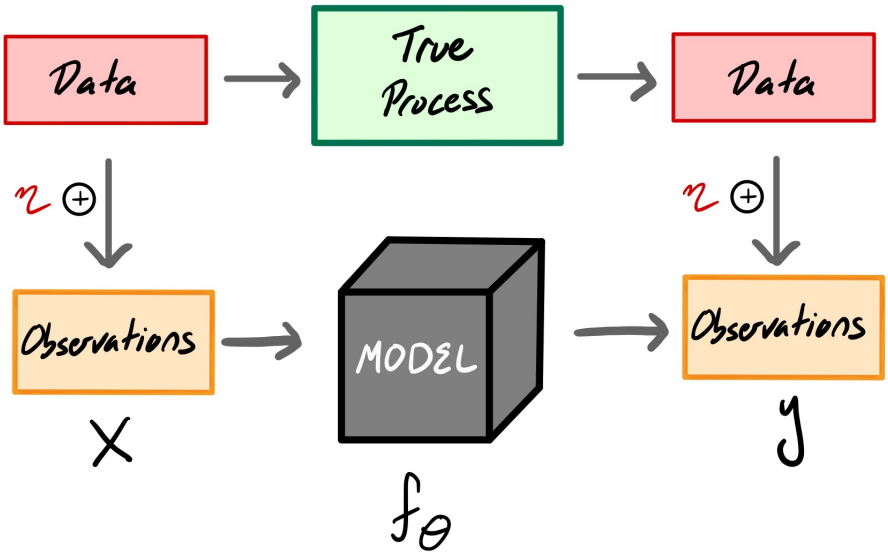


Figure A.1.: A schematic for the dichotomy between a model and a real process and how they are connected through observations. Our model, f, and the associated parameters, θ, is an approximate description of the true process which relates our observations. In physical models, these are often parameters of a system whereas in machine learning, these are weights for the function. Ultimately, our data, $\mathbf{x}, \mathbf{y}$, are based on observations in the form of measurements which are often noisy and incomplete representations. Overall, we need to characterize all sources of uncertainty which consists of the inputs, $\mathbf{x}$, the function, f, the function parameters, θ, and the outputs, $\mathbf{y}$.

It is important to clarify terminology when discussing definitions across disciplines so that the concepts are clear for both sides. Just like the definition of uncertainty characterization, there is some overlap in the definitions in the alleged sources of uncertainty and therefore there is some debate within the community about the precise meanings. Below we outline the sources of uncertainty in general terms used in the physical science community and within each of the definitions we will give the appropriate typical machine learning term that is found in the literature.

Measurement Uncertainty

This is the uncertainty associated with the data. This comes in the form of measurements from the phenomena of interest for example from the field or through remote sensing

platforms. In all cases, we will obtain the data via measurements which are inevitably noisy due to instrument error or mispecification which occurs even despite the improvement of our measuring capabilities over the last decade. In machine learning, we often refer to this as **aleatoric uncertainty** and we generally refer to this source of error as irreducible. It is assumed that the data is inherently stochastic and thus collecting more data for our models will not reduce errors we obtain from our models.

When modeling, we often distinguish the input data and the output data. In experiments, the input data is typically the free or fixed variable which is carefully controlled with a defined space of parameters. The output data are typically the observations where we try to predict a variable of interest. In machine learning we often do the same. We often expect some amount of irreducible error that will always occur when we compare our model predictions and the real data. To further improve our models, one can also distinguish a level of noise that is independent on the inputs (homoskedastic) or that is dependent of the inputs (heteroskedastic). In deterministic models, the homoskedastic assumption is the typical approach whereas in probabilistic models, both methods are seen in the literature.

However, one must be careful to distinguish this as actually inherent variation or just a relationship that we don't or cannot take into account. For example, it could be the case that we have samples that are not representative of the phenomena, in which case this is more of an experimental error and not a data uncertainty. Another example is the case where we choose to represent our observations in the form of a probability distribution to explain the data. But now this is a modeling decision via a probability distribution which adds another layer of uncertainty in the form of the model and not the actual data. There is even a debate stating that the concept of aleatoric uncertainty doesn't actual exist [Nearing et al., 2016, Heße et al., 2019] and that it's simply a necessary condition imposed via the Bayesian framework. In ML, this is imposed to acknowledge the inherent stochasticity even though there is no formal way to account for it. But what is clear is when we obtain input data from secondary sources (not truly raw data), this implies it is necessary to propagate that uncertainty through our subsequent model.

Model Uncertainty

Structure Uncertainty are the errors associated with our decisions for the function type f. All models are approximations of reality so this function will never completely represent the real system and therefore we can always expect residual error. However, we make conscious decisions about the form of f and its complexity. In very complex models describing our climate system, we have many complete systems to describe the atmosphere, land and ocean with different initial conditions, models of internal and external forcings, and a description of the dynamics [Reichstein et al., 2019]. One has to choose which processes and subprocesses describe the system and at what cost. A complex f is more precise but it is more computationally costly whereas a simpler f is more crude and approximate but more computationally efficient. *Parameter Uncertainty* occurs with

all of the parameters associated with the function f. These come in the form of assumptions and/or constraints, e.g. initial conditions and boundary conditions, which lead to different simulated outcomes. If the parameters are not fixed then one could possibly have a set of possible parameters that can effectively model the system and give adequate predictions especially in a stochastic scenario.

In machine learning, we categorize both function and parameter uncertainty as **epistemic uncertainty**. We formulate a hypothesis, $\mathcal{H}$ which consists of possible forms of f with parameters θ that can explain the relationship between datasets. From assumptions or prior experience, one has to choose from the many class of models that exist in the literature where each model type is a hypothesis, e.g. a simple model like linear regression, a set of basis functions, or a more flexible model like random forest regression. Very flexible models such as neural networks are currently the most popular methods for modeling and have a large set of parameters. They suffer from the same issue of different parameter initial conditions or different neural network structures can give good solutions. This uncertainty characterization is made explicit in Bayesian methods and the treatment of each parameter set also separates probabilistic models from discriminative models.

Scenario Uncertainty

This is perhaps the hardest uncertainty to deal with compared to the previous ones mentioned. Scenario uncertainty occurs when we attempt to model scenarios outside of our observations. Concretely, the datasets used to validate and calibrate our models are from a distribution $\mathbf{x} \sim \mathbb{P}$ yet any new dataset is from a different distribution $\mathbf{x}' \sim \mathbb{Q}$ which is distinct enough s.t. $\mathbb{P} \neq \mathbb{Q}$. The best example is in the context of forecasting, e.g. when we try to predict future global average temperature taking into account of unmodeled and unknown influence of anthropogenic contributions [Intergovernmental Panel on Climate Change. Working Group 1 et al., 2007]. This is a difficult problem because we are trying to predict something within a dynamic, evolving system for which we have no observations. Physical models are typically designed to span a range of possibilities but not necessarily predict them [Vuuren et al., 2011].

In machine learning terms, we refer to this as out-of-sample uncertainty via a *distribution-shift*. This is when the observations we use to train our ML models change and the models have difficulty adjusting to the changes. Just like Earth sciences, this is the hardest form of uncertainty to deal with as it is an inevitable portion of our datasets. This is related to the notion of generalization as we typically criticize many ML models for not learning/generalizing outside of the training datasets [Barbiero et al., 2020, Zhang et al., 2017] or are subject to adversarial attacks [Kurakin et al., 2017]. If no underlying function or pattern is assumed [Wilson and Adams, 2013] then ML models have issues with extrapolation. Even out best, most flexible machine learning methods, including fully probabilistic, all have issues with extrapolation [Maddox et al., 2019b].

So as shown by all of the sources of uncertainty, it is very clear there is a lot of take into account when considering all of the sources of uncertainty. Broadly speaking, there are

two main types of uncertainty quantification: forward and inverse. *Forward UQ* involves propagating the input uncertainties through our model such that they are reflected in the output uncertainty. We often assume we can summarize our inputs $\mathbf{x}$ by some probability distribution $\mathbb{P}$ and ideally we should see some indication that the confidence of the model predictions is dependent upon the uncertainty of the inputs. This will give the user some indication of reliability and performance of a system wrt to the data and this can lead to better subsequent decisions. *Inverse UQ* is when we consider some observations $\mathcal{D}$ and we want to see what the uncertainty is within the model parameters. This is often related to the parameters when we wish to discover or uncover models based on the data.

So this leads us to our fundamental question: *if we know that there some latent error in our input data, how can we account for this without our machine learning models?* In the next sections 1.2.1- 1.2.3 we outline the three modeling scenarios found in machine learning with special attention to how they deal with aleatoric and epistemic uncertainty.

B. Uncertain Inputs in Gaussian Processes

Contents

B.1. Gaussian Processes

Consider the regression setting where we assume the following model:

$$y = \mathbf{f}(\mathbf{x}) + \epsilon \tag{B.1}$$

where $\mathbf{x}$ is a discriminate vector of inputs, $\mathbf{f}(\cdot) = [f_1, \ldots, f_N]$ is a latent GP function, and $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$ is a independently, identically distributed (i.i.d.) Gaussian noise parameter. We place a GP **prior** for $p(\mathbf{f})$ s.t.

$$p(\mathbf{f}|\mathbf{X}, \theta) \sim \mathcal{GP}\left(\mathbf{m}_\theta, \mathbf{K}_\theta\right), \tag{B.2}$$

where $\mathbf{m}_\theta$ and $\mathbf{K}_\theta$ are the mean and covariance function for the GP, θ are the parameters of the model and $\mathbf{X}$ is the data. Combining this prior with the regression problem model from eq: B.1, we assume a likelihood function:

$$p(y|\mathbf{f}, \mathbf{X}) \sim \mathcal{N}(y|\mathbf{f}(\mathbf{x}), \sigma_y^2\mathbf{I}) \tag{B.3}$$

We can invoke Bayes rule giving us the joint posterior distribution:

$$p(\mathbf{f}, \mathbf{f}_*|y) = \frac{p(\mathbf{f}, \mathbf{f}_*)p(y|\mathbf{f})}{p(y)} \tag{B.4}$$

where $p(y)$ is the marginal likelihood which we can obtain by integrating out the latent variables $\mathbf{f}$:

$$p(y) = \int_{\mathbf{f}} p(y, \mathbf{f}) d\mathbf{f} \tag{B.5}$$

$$= \mathcal{N}(y|\mathbf{m}_\theta, \mathbf{K}_\theta + \sigma^2\mathbf{I}) \tag{B.6}$$

In a regression setting, we are more interested in predictions; given some parameters and some data, what is the predictive function $\mathbf{f}$? This is known as the posterior distribution:

$$p(\mathbf{f}|y) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathcal{GP}}, \boldsymbol{\nu}^2_{GP}) \tag{B.7}$$

Note that the full probability distribution should be $p(\mathbf{f}|y, \mathbf{X}, \theta)$ but we have omitted it for brevity.

Inference

First, given the joint distribution of $\mathbf{f}, \mathbf{f}_*$ conditioned on $\mathbf{X}, \mathbf{X}_*$

$$p(\mathbf{f}, \mathbf{f}_*|\mathbf{X}, \mathbf{X}_*) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_*\end{bmatrix}; \begin{bmatrix}\mathbf{m}_\theta(\mathbf{X})\\ \mathbf{m}_\theta(\mathbf{X}_*)\end{bmatrix}, \begin{bmatrix}\mathbf{K} & \mathbf{K}_*\\ \mathbf{K}_* & \mathbf{K}_{**}\end{bmatrix}\right) \tag{B.8}$$

If we condition on our training inputs $D = (\mathbf{X}, y)$, we can come up with a predictive distribution for test points $\mathbf{x}_*$ via

$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}(\boldsymbol{\mu}_{\mathcal{GP}*}, \boldsymbol{\nu}^2_{\mathcal{GP}**}) \tag{B.9}$$

and we can give the GP predictive mean and variance functions as

$$\boldsymbol{\mu}_{\mathcal{GP}} = \underbrace{m(\mathbf{x}_*)}_{\text{Prior Mean}} + \underbrace{\boldsymbol{k}_*\mathbf{K}^{-1}}_{\text{Kalman Gain}} \underbrace{(y - m(\mathbf{X}))}_{\text{Error}} \tag{B.10}$$

$$\boldsymbol{\nu}^2_{\mathcal{GP}} = k_{**} - \boldsymbol{k}_*\mathbf{K}^{-1}\boldsymbol{k}_*^\top. \tag{B.11}$$

If we integrate out the $\mathbf{f}$ (or just take the conditional distribution of the joint PDF), then we get:

$$p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, y) = \int_{\mathbf{f}} p(\mathbf{f}|\mathbf{X}, y)p(\mathbf{f}_*|\mathbf{X}_*, y)d\mathbf{f} \tag{B.12}$$

$$= \mathcal{N}(\mathbf{f}_*|\mu_*, \boldsymbol{\Sigma}_*) \tag{B.13}$$

and the joint distribution of $\mathbf{f}_*$ and unobserved y:

$$p(y, \mathbf{f}_*|\mathbf{X}, \mathbf{X}_*) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_*\end{bmatrix}; \begin{bmatrix}\mathbf{m}_\theta(\mathbf{X})\\ \mathbf{m}_\theta(\mathbf{X}_*)\end{bmatrix}, \begin{bmatrix}\mathbf{K}_\theta(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I} & \mathbf{K}_\theta(\mathbf{X}, \mathbf{X}_*)\\ \mathbf{K}_\theta(\mathbf{X}, \mathbf{X}_*) & \mathbf{K}_\theta(\mathbf{X}_*, \mathbf{X}_*)\end{bmatrix}\right) \tag{B.14}$$

which gives us the mean predictions and the variance in our predictions:

$$\boldsymbol{\mu}_{\mathcal{GP}} = \underbrace{m(\mathbf{x}_*)}_{\text{Prior Mean}} + \underbrace{\mathbf{k}_*\mathbf{K}^{-1}}_{\text{Kalman Gain}} \underbrace{(y - (\mathbf{X}))}_{\text{Error}} = m(\mathbf{x}_*) + \mathbf{K}_*\alpha \tag{B.15}$$

$$\boldsymbol{\nu}^2_{\mathcal{GP}} = \underbrace{k_{**}}_{\text{Prior Variance}} - \mathbf{k}_*\mathbf{K}^{-1}_{\mathcal{GP}}\mathbf{k}_*^\top \tag{B.16}$$

where $\alpha = \mathbf{K}^{-1}(y - m(\mathbf{X}))$ and $\mathbf{K}_{\mathcal{GP}} = \mathbf{K}_\theta(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}$. This is the typical formulation 1.3 which assumes that the output of $\mathbf{x}$ (and $\mathbf{x}_*$) is deterministic. In section 3, we will look at the case where $\mathbf{x}_*$ is stochastic.

GP Training

In GP model inference, one maximizes the likelihood of the data D given the hyper-parameters $\boldsymbol{\theta}, \sigma_y^2$. The marginal likelihood is given by:

$$p(y|\mathbf{X}, \theta) = \mathcal{N}\left(y|\mathbf{m}_\theta, \mathbf{K}_\theta + \sigma_y^2 \mathbf{I}\right) \tag{B.17}$$

We can find the hyper-parameters $\boldsymbol{\theta}$ by maximizing the marginal log-likelihood. So fully expanding of the eq: B.17, we get:

$$\log p(y|\mathbf{X}, \theta) = \underbrace{-\frac{1}{2}(y - \mathbf{m}_\theta)^\top \mathbf{K}_{\mathcal{GP}}^{-1}(y - \mathbf{m}_\theta)}_{\text{Data-Fit}} \underbrace{-\frac{1}{2}\log|\mathbf{K}_{GP}|}_{\text{Complexity}} - \frac{N}{2}\log 2\pi \tag{B.18}$$

This maximization automatically embodies Occam's razor which does a trade-off between model complexity and overfitting. This is closed form for all GPs but these days, we typically use automatic differentiation toolboxes to alleviate some of the burden. Irregardless, the two most expensive calculations are within this procedure as the inversion of $\mathbf{K}_{\mathcal{GP}}^{-1}$ and the $|\mathbf{K}_{\mathcal{GP}}|$; since $\mathbf{K} \in \mathbb{R}^{N \times N}$, then these calculations are $\mathcal{O}(N^3)$ in operations and $\mathcal{O}(N^2)$ in memory costs. The kernel function is one of the most important aspects within the GP training regime. Once the kernel has been chosen to best reflect the problem at hand it has been found in Chen and Wang [2018] that any prior over the hyper-parameters does not provide significant improvements in GP predictions. However, note that the community is notorious for using the isotropic RBF kernel by default when conducting research. This kernel is the most flexible among the kernel family but not necessarily the most expressive Krauth et al. [2017].

B.1.1. Drawbacks

> "*It is important to keep in mind that Gaussian processes are not appropriate priors for all problems.*"
> – Neal, 1998

It is important to note that although the GP algorithm is one of the most trusted and reliable algorithms, it is not always the best algorithm to use for all problems. Below we mention a few drawbacks that the standard GP algorithm has along with some of the standard approaches to overcoming these drawbacks.

Gaussian Marginals. GPs have problems modeling heavy-tailed, asymmetric or multi-modal marginal distributions. There are some methods that change the likelihood so that it is heavy tailed [Jylänki et al., 2011, Shah et al., 2014] but this would remove the conjugacy of the likelihood term which would incur difficulties during fitting. Deep GPs and latent covariate models are an improvement to this limitation. A very popular approach is to construct a fully Bayesian model. This entails hyperpriors over the kernel parameters and Monte carlo sampling methods such as Gibbs sampling [Titsias et al., 2008], slice

sampling [Murray and Adams, 2010], Hamiltonian Monte Carlo [de G. Matthews et al., 2018a], and Sequential Monte Carlo [Svensson et al., 2015]. These techniques will capture more complex distributions. With the advent of better software [Salvatier et al., 2016, Phan et al., 2019] and more advanced sampling techniques like a differentiable iterative NUTS implementation [Phan et al., 2019], the usefulness of MC schemes is resurfacing.

Limited Number of Moments. This is related to the previous limitation: the idea that an entire function can be captured in terms of two moments: a mean and a covariance. There are some relationships which are difficult to capture without an adequate description, e.g. discontinuities [Neal, 1996] and non-stationary processes, and thus is a limitation of the GP priors we choose. The advent of warping the inputs or outputs of a GP has becoming a very popular technique to deal with the limited expressivity of kernels. Input warping is popular in methods such as deep kernel learning whereby a Neural network is used to capture the features and are used as inputs to the kernel function output warping is common in chained [Saul et al., 2016] and heteroscedastic methods where the function output is warped by another GP to capture the noise model of the data. Deep Gaussian processes [Damianou, 2015] can be thought of input and output warping methods due the multi-layer composition of function inputs and outputs.

Linearity of Predictive Mean. The predictive mean of a GP is linear to the observations, i.e. $\mu_{GP} = \mathbf{K}\alpha$. This essentially is a smoother which can be very powerful but also will miss key features. If there is some complex structured embedded within the dataset, then a GP model can never really capture this irregardless of the covariance function found.

Predictive Covariance. The GP predictive variance is a function of the training inputs and it is independent of the observed inputs. This is important if the input data has some information which could be used to help determine the regions of uncertainty, e.g. the gradient. An example would be data on a spatial grid whereby some regions points would have more certainty than others which could be obtained by knowing the input location and not necessarily the expected output.

B.2. Sparse Gaussian Processes

One big issue with the standard GP formulation (Section B) is that the inverse of the $\mathbf{K}_{\mathcal{GP}}^{-1}$ is $\mathcal{O}(N^3)$ which can be very expensive. One can use inducing points which act as a subset of points M where M is much less than N, $M << N$. This can be used to reduce the computation to a cost of $\mathcal{O}(NM^2)$. There are a number of different methods using this idea including methods like subset of regressors, Fully Independent Training Conditional, and sparse variational Gaussian processes [Candela and Rasmussen, 2005, Bui et al., 2017b, Hensman et al., 2015]. In this thesis, we focus on a particular implementation called sparse variational free energy (VFE) method [Titsias and Lawrence, 2010]. This performs an approximate inference scheme by introducing a variational parameter $q(f)$ over the latent function. Then, we can optimize a lower bound on the likelihood (ELBO) to approximate the posterior.

$$\log p(y) \geq \log p(y) - \mathrm{KL}\left[q(\mathbf{f})||p(\mathbf{f}|y)\right] \tag{B.19}$$

$$\geq \log \mathcal{N}(y; 0, \mathbf{Q}_{\mathbf{ff}} + \sigma^2\mathbf{I}) - \frac{1}{2\sigma^2}\mathrm{Tr}\left(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}\right) \tag{B.20}$$

where $\mathbf{Q}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{f}u}\mathbf{K}_{uu}^{-1}\mathbf{K}_{u\mathbf{f}}$ is the Nystrom approximation of $\mathbf{K}_{\mathbf{ff}}$ and u is a small subset of $M << N$ inducing points at locations $\{z_j\}_{j=1}^M$ which makes $[\mathbf{K}_{\mathbf{f}u}]_{ij} = k(\mathbf{x}_i, \mathbf{z}_j)$ and $[\mathbf{K}_{uu}]_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$. The first term of the ELBO corresponds to a deterministic training conditional (DTC) [Candela and Rasmussen, 2005] and the added regularization trace term prevents overfitting which has is a problem with the DTC. Since this variational approximation is a Gaussian distribution, $q(y_*) = \mathcal{N}(y_*; \mu_*, \sigma_*^2)$, there is a closed-form predictive mean and variance given by:

$$\mu_{\mathcal{SGP}}(\mathbf{x}_*) = \mathbf{k}_{u*}^\top \left(\mathbf{K}_{u\mathbf{f}}\mathbf{K}_{\mathbf{f}u} + \sigma^2\mathbf{K}_{uu}\right)^{-1}\mathbf{K}_{u\mathbf{f}}\, y \tag{B.21}$$

$$\sigma^2_{\mathcal{SGP}}(\mathbf{x}_*) = \sigma^2 + k_{**} - \mathbf{k}_{u*}^\top\mathbf{K}_{uu}^\top\mathbf{k}_{u*} + \mathbf{k}_{u*}^\top\left(\sigma^{-2}\mathbf{K}_{u\mathbf{f}}\mathbf{K}_{\mathbf{f}u} + \mathbf{K}_{uu}\right)\mathbf{k}_{u*} \tag{B.22}$$

B.3. Analytic Moments

The posterior of this distribution is non-Gaussian because we have to propagate a probability distribution through a non-linear kernel function. So this integral becomes intractable. We can compute the analytical Gaussian approximation by only computing the mean and the variance of the

Mean Function

$$
\begin{aligned}
m(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) &= \mathbb{E}_{\mathbf{f}_*}\left[f_* \, \mathbb{E}_{\mathbf{x}_*}\left[p(f_*|\mathbf{x}_*)\right]\right] && \text{(B.23)}\\
&= \mathbb{E}_{\mathbf{x}_*}\left[\mathbb{E}_{f_*}\left[f_* \, p(f_*|\mathbf{x}_*)\right]\right] && \text{(B.24)}\\
&= \mathbb{E}_{x_*}\left[\mu_{\mathrm{GP}}(\mathbf{x}_*)\right] && \text{(B.25)}
\end{aligned}
$$

Variance Function

The variance term is a bit more complex.

$$
\begin{aligned}
v(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) &= \mathbb{E}_{\mathbf{f}_*}\left[f_*^2 \, \mathbb{E}_{\mathbf{x}_*}\left[p(f_*|\mathbf{x}_*)\right]\right] - \left(\mathbb{E}_{\mathbf{f}_*}\left[f_* \, \mathbb{E}_{\mathbf{x}_*}\left[p(f_*|\mathbf{x}_*)\right]\right]\right)^2 && \text{(B.26)}\\
&= \mathbb{E}_{\mathbf{x}_*}\left[\mathbb{E}_{\mathbf{x}_*}\left[f_*^2 \, p(f_*|\mathbf{x}_*)\right]\right] - \left(\mathbb{E}_{\mathbf{x}_*}\left[\mathbb{E}_{\mathbf{x}_*}\left[f_* \, p(f_*|\mathbf{x}_*)\right]\right]\right)^2 && \text{(B.27)}\\
&= \mathbb{E}_{\mathbf{x}_*}\left[\sigma^2_{\mathrm{GP}}(\mathbf{x}_*) + \mu^2_{\mathrm{GP}}(\mathbf{x}_*)\right] - \mathbb{E}_{x_*}\left[\mu_{\mathrm{GP}}(\mathbf{x}_*)\right]^2 && \text{(B.28)}\\
&= \mathbb{E}_{\mathbf{x}_*}\left[\sigma^2_{\mathrm{GP}}(\mathbf{x}_*)\right] + \mathbb{E}_{\mathbf{x}_*}\left[\mu^2_{\mathrm{GP}}(\mathbf{x}_*)\right] - \mathbb{E}_{x_*}\left[\mu_{\mathrm{GP}}(\mathbf{x}_*)\right]^2 && \text{(B.29)}\\
&= \mathbb{E}_{\mathbf{x}_*}\left[\sigma^2_{\mathrm{GP}}(\mathbf{x}_*)\right] + \mathbb{V}_{\mathbf{x}_*}\left[\mu_{\mathrm{GP}}(\mathbf{x}_*)\right] && \text{(B.30)}
\end{aligned}
$$

B.4. Taylor Approximation Derivation

We will approximate our mean and variance function via a Taylor Expansion. First let's take a step back and look at the taylor expansion of a single function f w.r.t. to $\mathbf{x}_*$ which is characterized by its mean $\mu_{\mathbf{x}_*}$ and variance function $\Sigma_{\mathbf{x}_*}$. Taking the first two orders, we get

$$\mathbf{z}_\mu = f(\mathbf{x}_*) \approx \underbrace{f(\mu_{\mathbf{x}_*})}_{\text{Zeroth Order}} + \underbrace{\nabla_{\mathbf{x}_*} f\Big|_{\mathbf{x}_*=\mu_{\mathbf{x}}} (\mathbf{x}_* - \mu_{\mathbf{x}_*})}_{\text{1st Order}} \tag{B.31}$$

$$+ \underbrace{\nabla^2_{\mathbf{x}_*} f\Big|_{\mathbf{x}_*=\mu_{\mathbf{x}}} (\mathbf{x}_* - \mu_{\mathbf{x}_*})^\top (\mathbf{x}_* - \mu_{\mathbf{x}_*})}_{\text{2nd Order}} + \underbrace{\mathcal{O}({\mathbf{x}_*}^3)}_{\text{Higher Order}} \tag{B.32}$$

Mean Function

Now we need to take the expectation of our approximation, $\mathbb{E}[\mathbf{z}_\mu]$. We tackle each of the terms individually below.

Zeroth Term

For the first term, we take the expectation.

$$\mathbb{E}_{\mathbf{x}_*}\left[f(\mu_{\mathbf{x}_*})\right] = f(\mu_{\mathbf{x}_*})$$

This is the same because the expectation of the mean of a function f is simply the function f evaluated at the mean.

1st Order

The 1st order term is given by:

$$\mathbb{E}_{\mathbf{x}_*}\left[\nabla_{\mathbf{x}_*} f\Big|_{\mathbf{x}_*=\mu_{\mathbf{x}}} (\mathbf{x}_* - \mu_{\mathbf{x}_*})\right] = \nabla_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*}) \mathbb{E}_{\mathbf{x}_*}\left[(\mathbf{x}_* - \mu_{\mathbf{x}_*})\right] = 0 \tag{B.33}$$

$\mathbb{E}[(\mathbf{x}_* - \mu_{\mathbf{x}_*})] = 0$ because the terms cancel each other out.

2nd Order

The 2nd order term is given by:

$$\mathbb{E}_{\mathbf{x}_*}\left[\nabla^2_{\mathbf{x}_*} f\Big|_{\mathbf{x}_*=\mu_{\mathbf{x}}} (\mathbf{x}_* - \mu_{\mathbf{x}_*})^\top (\mathbf{x}_* - \mu_{\mathbf{x}_*})\right] = \nabla^2_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*}) \mathbb{E}_{\mathbf{x}_*}\left[(\mathbf{x}_* - \mu_{\mathbf{x}_*})^\top (\mathbf{x}_* - \mu_{\mathbf{x}_*})\right] \tag{B.34}$$

We have the covariance term so we can simplify this:

$$\mathbb{E}_{\mathbf{x}_*}\left[\nabla^2_{\mathbf{x}_*} f\Big|_{\mathbf{x}_*=\mu_{\mathbf{x}}} (\mathbf{x}_* - \mu_{\mathbf{x}_*})^\top (\mathbf{x}_* - \mu_{\mathbf{x}_*})\right] = \nabla^2_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})\Sigma_{\mathbf{x}_*}$$

So we're left with:

$$\mathbb{E}_{\mathbf{x}_*}\left[f(\mathbf{x}_*)\right] \approx f(\mu_{\mathbf{x}_*}) + \nabla^2_{\mathbf{x}_*} f(\mu_{\mathbf{x}})^\top \Sigma_{\mathbf{x}_*} + \mathcal{O}({\mathbf{x}_*}^3)$$

which we can simplify to:

$$f(\mathbf{x}_*) \approx f(\mu_{\mathbf{x}_*}) + \underbrace{\frac{1}{2}\text{Tr}\left\{\nabla^2_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*}\right\}}_{\text{2nd Order}}$$

Linearized GP Mean

So now instead of a simple function f, we have our GP predictive mean equation μ_{GP} which we can simply plug into the approximation.

$$\tilde{\mu}_{\text{LinGP}}(\mathbf{x}_*) = \mu_{\text{GP}}(\mu_{\mathbf{x}_*}) + \underbrace{\frac{1}{2}\text{Tr}\left\{\nabla^2_{\mathbf{x}_*} \mu_{\text{GP}}(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*}\right\}}_{\text{2nd Order}}$$

Variance Function

So the variance term is a bit more difficult to calculate due to the $\mathbb{V}[\cdot]$ operator.

$$\tilde{\sigma}^2_{\text{LinGP}}(\mathbf{x}_*) = \mathbb{E}_{\mathbf{x}_*}\left[\sigma^2_{\text{GP}}(\mathbf{x}_*)\right] + \mathbb{V}_{\mathbf{x}_*}\left[\mu_{\text{GP}}(\mathbf{x}_*)\right]$$

Term I. The formulation for the expectation of the Taylor expanded predictive variance function is similar to the equation above. So we can replace μ_{GP} with σ^2_{GP}.

$$\mathbb{E}_{\mathbf{x}_*}\left[\sigma^2_{\text{GP}}(\mathbf{x}_*)\right] = \sigma^2_{\text{GP}}(\mu_{\mathbf{x}_*}) + \underbrace{\frac{1}{2}\text{Tr}\left\{\nabla^2_{\mathbf{x}_*} \sigma^2_{\text{GP}}(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*}\right\}}_{\text{2nd Order}}$$

Term II. This term is more difficult to calculate. Again, we'll take a step back and see how this is for a function f. If we want a first order approximation, we will have the following:

$$\mathbb{V}_{\mathbf{x}_*}\left[f(\mathbf{x}_*)\right] = \nabla_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*} \nabla_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})$$

This is a sufficient approximation when $f(\mathbf{x})$ is approximately linear and/or when $\Sigma_{\mathbf{x}_*}$ is relatively small compared to $f(\mu_{\mathbf{x}_*})$. Alternatively, we can add a second order approxi-

mation which would add the following terms:

$$\mathbb{V}_{\mathbf{x}_*}\left[f(\mathbf{x}_*)\right] \approx \underbrace{\left(\nabla_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})\right)^2 \Sigma_{\mathbf{x}_*}}_{\text{1st Order}} \tag{B.35}$$

$$\underbrace{-\frac{1}{4}\left(\nabla^2_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})\right)^2 \Sigma_{\mathbf{x}_*} + \mathbb{E}_{\mathbf{x}_*}\left[\mathbf{x}_* - \mu_{\mathbf{x}_*}\right] \nabla^3_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*}) + \frac{1}{4}\mathbb{E}_{\mathbf{x}_*}\left[\mathbf{x}_* - \mu_{\mathbf{x}_*}\right]\left(\nabla^2_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})\right)^2}_{\text{2nd Order}} \tag{B.36}$$

This expression has 3rd and 4th central moments with respect to the mean. These terms are often negligible according to the conditions mentioned above. In addition, it is a very expensive calculation for some functions. So a practical compromise is to use the 2nd order approximation for the mean and the first order approximation for the variance. This is the approach given here as 3rd and 4th central moments of kernel functions is very expensive. So combining the terms together, we get:

$$\mathbb{V}_{\mathbf{x}_*}\left[f(\mathbf{x}_*)\right] \approx f(\mu_{\mathbf{x}_*}) + \frac{1}{2}\text{Tr}\left\{\nabla^2_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*}\right\} + \left(\nabla_{\mathbf{x}_*} f(\mu_{\mathbf{x}_*})\right)^2 \Sigma_{\mathbf{x}_*} \tag{B.37}$$

So then subsituting all of the portions for the appropriate GP function, we get the following:

$$\tilde{\sigma}^2_{\text{LinGP}}(\mathbf{x}_*) = \sigma^2_{\text{GP}}(\mu_{\mathbf{x}_*}) + \left(\nabla_{\mathbf{x}_*}\mu_{\text{GP}}(\mu_{\mathbf{x}_*})\right)^2 \Sigma_{\mathbf{x}_*} + \frac{1}{2}\text{Tr}\left\{\nabla^2_{\mathbf{x}_*}\sigma^2_{\text{GP}}(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*}\right\} \tag{B.38}$$

Linearized Predictive Mean and Variance

$$\tilde{\mu}_{\text{LinGP}}(\mathbf{x}_*) = \underbrace{\mu_{\text{GP}}(\mu_{\mathbf{x}_*})}_{\text{1st Order}} + \underbrace{\frac{1}{2}\text{Tr}\left\{\nabla^2_{\mathbf{x}_*}\mu_{\text{GP}}(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*}\right\}}_{\text{2nd Order}} \tag{B.39}$$

$$\tilde{\sigma}^2_{\text{LinGP}}(\mathbf{x}_*) = \underbrace{\sigma^2_{\text{GP}}(\mu_{\mathbf{x}_*}) + \nabla_{\mathbf{x}_*}\mu_{\text{GP}}(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*} \nabla_{\mathbf{x}_*}\mu_{\text{GP}}(\mu_{\mathbf{x}_*})}_{\text{1st Order}} \tag{B.40}$$

$$+ \underbrace{\frac{1}{2}\text{Tr}\left\{\nabla^2_{\mathbf{x}_*}\sigma^2_{\text{GP}}(\mu_{\mathbf{x}_*})^\top \Sigma_{\mathbf{x}_*}\right\}}_{\text{2nd Order}} \tag{B.41}$$

where $\nabla_{\mathbf{x}_*}$ is the gradient of the function w.r.t. $\mathbf{x}$ and $\nabla^2_{\mathbf{x}_*}$ is the second derivative (the Hessian) of the function w.r.t. $\mathbf{x}_*$. This is a second-order approximation which has that expensive Hessian term. There have have been studies that have shown that that term tends to be neglible in practice and a first-order approximation is typically enough.

Practically speaking, this leaves us with the following predictive mean and variance functions:

$$\mu_{\text{GP}}(\mathbf{x}_*) = k(\mathbf{x}_*)\,\mathbf{K}^{-1}_{GP} y = k(\mathbf{x}_*)\,\alpha \tag{B.42}$$

$$\nu^2_{GP}(\mathbf{x}_*) = \sigma^2_y + \nabla_{\mu_{\text{GP}}}\,\Sigma_{\mathbf{x}_*}\,\nabla^\top_{\mu_{\text{GP}}} + k_{**} - \mathbf{k}_*(\mathbf{K} + \sigma^2_y \mathbf{I}_N)^{-1}\mathbf{k}^\top_* \tag{B.43}$$

As seen above, the only extra term we need to include is the derivative of the mean function that is present in the predictive variance term.

Sparse GPs

We can extend this method to other GP algorithms including sparse GP models. The only thing that changes are the original μ_{GP} and v^2_{GP} equations. In a sparse GP we have the following predictive functions

$$\mu_{SGP} = K_{*z}K_{zz}^{-1}m \tag{B.44}$$

$$v^2_{SGP} = K_{**} - K_{*z}\left[K_{zz}^{-1} - K_{zz}^{-1}SK_{zz}^{-1}\right]K_{*z}^{\top} \tag{B.45}$$

So the new predictive functions will be:

$$\mu_{SGP} = k_{*z}K_{zz}^{-1}m \tag{B.46}$$

$$v^2_{SGP} = K_{**} - K_{*z}\left[K_{zz}^{-1} - K_{zz}^{-1}SK_{zz}^{-1}\right]K_{*z}^{\top} + \tilde{\Sigma}_x \tag{B.47}$$

As shown above, this is a fairly extensible method that offers a cheap improved predictive variance estimates on an already trained GP model. Some future work could be evaluating how other GP models, e.g. Sparse Spectrum GP, Multi-Output GPs, e.t.c.

B.5. Moment Matching Derivation

Recall - The Law of Iterated Expectations and Conditional Variance

$$\mathbb{E}[y] = \mathbb{E}_x[\mathbb{E}[y|x]] \tag{B.48}$$
$$\mathbb{V}[y] = \mathbb{V}_x[\mathbb{E}[y|x]] + \mathbb{E}_x[\mathbb{V}[y|x]] \tag{B.49}$$

Predictive Mean and Variance

So for the GP predictive mean and variance, we can apply the same formula. This is equivalent to computing the first and second order central moments of the corresponding equations. This is useful for us because we know the output distribution is non-Gaussian but we can approximate it to be Gaussian by computing the mean and variance of the GP predictive mean and GP predictive variance wrt to the noisy inputs $\mathbf{x}$.

$$\tilde{\mu}_{GP}(\mathbf{x}_*) = \mathbb{E}_{\mathbf{x}_*}\left[\mathbb{E}_{f_*}[f_*|\mathbf{x}_*]\right] \tag{B.50}$$
$$= \mathbb{E}_{\mathbf{x}_*}\left[\mu_{GP}(\mathbf{x}_*)\right] \tag{B.51}$$
$$\tilde{\sigma}^2_{GP}(\mathbf{x}_*) = \mathbb{V}_{\mathbf{x}_*}\left[\mathbb{E}_{\mathbf{x}_*}[|\mathbf{x}_*]\right] + \mathbb{E}_{\mathbf{x}_*}[\mathbb{V}_{\mathbf{x}_*}[|\mathbf{x}_*]] \tag{B.52}$$
$$= \mathbb{V}_{\mathbf{x}_*}\left[\mu_{GP}(\mathbf{x}_*)\right] + \mathbb{E}_{\mathbf{x}_*}[\sigma^2_{GP}(\mathbf{x}_*)] \tag{B.53}$$
$$= \mathbb{E}_{\mathbf{x}_*}\left[\mu^2_{GP}(\mathbf{x}_*)\right] - \mathbb{E}^2_{\mathbf{x}_*}\left[\mu_{GP}(\mathbf{x}_*)\right] + \mathbb{E}_{\mathbf{x}_*}[\sigma^2_{GP}(\mathbf{x}_*)] \tag{B.54}$$

Mean Function

For this function, we

$$\begin{aligned}
\tilde{\mu}_{GP}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) &= \mathbb{E}_{\mathbf{x}_*}\left[\mu_{GP}(\mathbf{x}_*)\right] \\
&= \int_{\mathcal{X}} \left[m_{GP}(\mathbf{x}_*) + k(\mathbf{X}, \mathbf{x}_*)^\top \mathbf{K}_{GP}^{-1}(\mathbf{y} - m_{GP}(\mathbf{x}_*))\right]\ p(\mathbf{x}_*) d\mathbf{x}_* \\
&= \int_{\mathcal{X}} m_{GP}(\mathbf{x}_*)\ p(\mathbf{x}_*)\ d\mathbf{x}_* + \int_{\mathcal{X}} k(\mathbf{X}, \mathbf{x}_*)\mathbf{K}_{\mathcal{GP}}^{-1}(\mathbf{y} - m_{\mathcal{GP}}(\mathbf{x}_*))\ p(\mathbf{x}_*) d\mathbf{x}_* \\
&= \int_{\mathcal{X}} k(\mathbf{X}, \mathbf{x}_*) \underbrace{\mathbf{K}_{\mathcal{GP}}^{-1}\mathbf{y}}_{\alpha}\ p(\mathbf{x}_*) d\mathbf{x}_* \\
&= \int_{\mathcal{X}} k(\mathbf{X}, \mathbf{x}_*)^\top \alpha\ p(\mathbf{x}_*)\ d\mathbf{x}_* \\
&= \alpha^\top \underbrace{\int_X k(\mathbf{X}, \mathbf{x}_*) \cdot p(\mathbf{x}_*) d\mathbf{x}_*}_{\Psi_1} \\
\tilde{\mu}_{GP}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) &= \Psi_1^\top \alpha
\end{aligned}$$

Predictive Variance

$$\tilde{\sigma}^2_{GP}(\mathbf{x}_*) = \underbrace{\mathbb{E}_{\mathbf{x}_*}[\sigma^2_{GP}(\mathbf{x}_*)]}_{\text{Term I}} + \underbrace{\mathbb{E}_{\mathbf{x}_*}[\mu^2_{GP}(\mathbf{x}_*)]}_{\text{Term II}} - \underbrace{\mathbb{E}_{\mathbf{x}_*}[\mu_{GP}(\mathbf{x}_*)]^2}_{\text{Term III}}$$

Term I

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}_*}[\sigma^2_{GP}(\mathbf{x}_*)] &= \int_{\mathcal{X}} \left[k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{X}, \mathbf{x}_*)\mathbf{K}^{-1}_{GP}k(\mathbf{X}, \mathbf{x}_*)^\top\right]\ p(\mathbf{x}_*)d\mathbf{x}_* \\
&= \int_{\mathcal{X}} k(\mathbf{x}_*, \mathbf{x}_*)\ p(\mathbf{x}_*)d\mathbf{x}_* - \int_{\mathcal{X}} k(\mathbf{X}, \mathbf{x}_*)\mathbf{K}^{-1}_{GP}k(\mathbf{X}, \mathbf{x}_*)^\top\ p(\mathbf{x}_*)d\mathbf{x}_* \\
&= \int_{\mathcal{X}} k(\mathbf{x}_*, \mathbf{x}_*)\ p(\mathbf{x}_*)d\mathbf{x}_* - \sum_{i,j} \mathbf{K}^{-1}_{GP(i,j)}k(\mathbf{x_i}, \mathbf{x}_*)k(\mathbf{x_j}, \mathbf{x}_*)\ p(\mathbf{x}_*) \\
&= \int_{\mathcal{X}} k(\mathbf{x}_*, \mathbf{x}_*)\ p(\mathbf{x}_*)d\mathbf{x}_* - \text{Tr}\left(\mathbf{K}^{-1}_{GP}\int_{\mathcal{X}} k(\mathbf{X}, \mathbf{x}_*)k(\mathbf{X}, \mathbf{x}_*)\ p(\mathbf{x}_*)d\mathbf{x}_*\right) \\
&= \psi_0 - \text{Tr}\left(\mathbf{K}^{-1}_{GP}\Psi_2\right)
\end{aligned}$$

Term II

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}_*}[\mu^2_{GP}(\mathbf{x}_*)] &= \int_{\mathcal{X}} k(\mathbf{X}, \mathbf{x}_*)^\top \alpha\alpha^\top k(\mathbf{X}, \mathbf{x}_*)\ p(\mathbf{x}_*)d\mathbf{x}_* \\
&= \sum_{i,j} \alpha_i\alpha_j k(\mathbf{x_i}, \mathbf{x}_*)k(\mathbf{x_j}, \mathbf{x}_*)\ p(\mathbf{x}_*) \\
&= \text{Tr}\left(\alpha\alpha^\top \int_{\mathcal{X}} k(\mathbf{X}, \mathbf{x}_*)\ k(\mathbf{X}, \mathbf{x}_*)\ p(\mathbf{x}_*)\ d\mathbf{x}_*\right) \\
&= \text{Tr}\left(\alpha\alpha^\top \Psi_2\right)
\end{aligned}$$

Term III

This is the squared expected value of the GP mean w.r.t. the noisy inputs $\mathbf{x}_*$. We've already calculated this above so we can just substitute this expression and square it:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}_*}[\mu_{GP}(\mathbf{x}_*)]^2 &= [\tilde{\mu}_{GP}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})]^2 \\
&= [\Psi_1^\top \alpha]^2 \\
&= \text{Tr}\left(\Psi_1\Psi_1^\top \alpha\alpha^\top\right)
\end{aligned}$$

Final Solution

$$\begin{aligned}
\tilde{\sigma}^2_{GP}(\mathbf{x}_*) &= \psi_0 - \text{Tr}\left(\mathbf{K}^{-1}_{GP}\Psi_2\right) + \text{Tr}\left(\alpha\alpha^\top\Psi_2\right) - \text{Tr}\left(\Psi_1\Psi_1^\top\alpha\alpha^\top\right) \\
&= \psi_0 - \text{Tr}\left(\left(\mathbf{K}^{-1}_{GP} - \alpha\alpha^\top\right)\Psi_2\right) - \text{Tr}\left(\Psi_1\Psi_1^\top\alpha\alpha^\top\right) \\
&= \psi_0 - \text{Tr}\left(\left(\mathbf{K}^{-1}_{GP} - \alpha\alpha^\top\right)\Psi_2 - \Psi_1\Psi_1^\top\alpha\alpha^\top\right) \\
&= \psi_0 - \text{Tr}\left(\left(\mathbf{K}^{-1}_{GP} - \alpha\alpha^\top\right)\Psi_2 - (\Psi_1^\top\alpha)^2\right)
\end{aligned}$$

C. Gaussianizing the Earth

Contents

C.1. Equivalence: KL-Divergence and Log-Likelihood

Here we want to show that the KL-Divergence between the true distribution $p_{\text{data}}(\mathbf{x})$ and the estimated distribution $p_\theta(\mathbf{x})$ is the same as maximizing the likelihood of our estimated distribution $p_\theta(\mathbf{x})$.

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_\theta(\mathbf{x})\right] = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] + \text{constant} \tag{C.1}$$

Proof

First we decompose the KL-Divergence into its log terms.

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_\theta(\mathbf{x})\right] &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log \frac{p_{\text{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\right] && \text{(C.2)}\\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_{\text{data}}(\mathbf{x}) - \log p_\theta(\mathbf{x})\right] && \text{(C.3)}\\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_{\text{data}}(\mathbf{x})\right] - \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] && \text{(C.4)}
\end{aligned}$$

The first term is the entropy of our data, $H\left(p_{\text{data}}(\mathbf{x})\right)$. This term doesn't depend on our parameters θ which means it will be constant irregardless of how well we estimate $p_\theta(\mathbf{x})$. So we can simplify this function.

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_\theta(\mathbf{x})\right] &= -\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] + \text{constant} && \text{(C.5)}\\
&= -\int p_{\text{data}}(\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{x} + C && \text{(C.6)}
\end{aligned}$$

The remaining term is the cross-entropy; the expected amount of bits need to compress. This is optimal when $p_{\text{data}}(\mathbf{x}) = p_\theta(\mathbf{x})$ (cite: Shannon Source Coding Theorem). Let $p_{\text{data}}(\mathbf{x})$ be an empirical distribution described by a delta.

$$p_{\text{data}}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \delta(\mathbf{x} - \mathbf{x}_n) \tag{C.7}$$

We assume that it puts a probability on the observed data and zero everywhere else. Plugging this into our KL-Divergence function, we get:

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_\theta(\mathbf{x})\right] = -\int \left[\frac{1}{N}\sum_{n=1}^{N}\delta(\mathbf{x}-\mathbf{x}_n)\right]\log p_\theta(\mathbf{x})d\mathbf{x} + C \tag{C.8}$$

Then using the law of large numbers where given enough samples we can empirically estimate this integral, we can simplify this even further:

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_\theta(\mathbf{x})\right] = -\frac{1}{N}\sum_{n=1}^{N}\log p_\theta(\mathbf{x}) + C \tag{C.9}$$

We are left with the log-likelihood term. So maximizing the likelihood of our estimated distribution $p_\theta(\mathbf{x})$ is equivalent to minimizing the difference between the estimated distribution $p_\theta(\mathbf{x})$ and the real distribution $p_{\text{data}}(\mathbf{x})$. This is a proxy method allowing us to find the parameters θ without explicitly knowing the real distribution.

C.2. Equivalence: Constructive-Destructive KL-Divergence

Let $\mathbf{f}_\theta$ be the invertible, bijective normalizing function which maps $\mathbf{x}$ to $\mathbf{z}$, i.e. $\mathbf{f}_\theta : \mathbf{x} \in \mathbb{R}^D \to \mathbf{z} \in \mathbb{R}^D$. Let g_θ be the inverse of $\mathbf{f}_\theta$ which is the generating function mapping $\mathbf{z}$ to $\mathbf{x}$, i.e. $\mathbf{g}_\theta := \mathbf{f}_\theta^{-1} : \mathbf{z} \in \mathbb{R}^D \to \mathbf{x} \in \mathbb{R}^D$. We can view $\mathbf{f}_\theta$ as a destructive density whereby we "destroy" the density of the original dataset $p_{\text{data}}(\mathbf{x})$ into a common base density $p_{\mathbf{z}}$. Conversely, we can view g_θ as a constructive density whereby we "construct" the density of the original dataset $p_{\text{data}}(\mathbf{x})$ from a base density $p_{\mathbf{z}}$.

$$\mathbf{z} = \mathbf{f}_\theta(\mathbf{x}), \qquad \mathbf{x} = g_\theta(\mathbf{z}) \tag{C.10}$$

We're assuming $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$. Using the change of variables formula, we can express the probability of $p_\theta(\mathbf{x})$ in terms of $\mathbf{z}$ and the transform $\mathbf{f}_\theta$.

$$p_\theta(\mathbf{x}) = p_{\mathbf{z}}(\mathbf{f}_\theta(\mathbf{x}))\left|\nabla_{\mathbf{x}}\mathbf{f}_\theta(\mathbf{x})\right| \tag{C.11}$$

This function $\mathbf{f}_\theta$ "normalizes" the complex density $\mathbf{x}$ into a simpler base distribution $\mathbf{z}$. We can also express this equation in terms of g_θ which is the standard found in the normalizing flow literature.

$$p_\theta(\mathbf{x}) = p_{\mathbf{z}}(\mathbf{z})\left|\nabla_{\mathbf{z}}g_\theta(\mathbf{z})\right|^{-1} \tag{C.12}$$

The function g_θ pushes forward the base density $\mathbf{z}$ to a more complex density $\mathbf{x}$. In this demonstration, we want to show that the following is equivalent.

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_{\mathbf{x}}(\mathbf{x};\theta)\right] = \mathrm{D}_{\mathrm{KL}}\left[p_{\text{target}}(\mathbf{z};\theta)||p_{\mathbf{z}}(\mathbf{z})\right] \tag{C.13}$$

This says that the KL-Divergence between the data distribution $p_{\text{data}}(\mathbf{x})$ and the model $p_{\mathbf{x}}(\mathbf{x};\theta)$ is equivalent to the KL-Divergence between *induced* distribution $p_{\text{target}}(\mathbf{z};\theta)$ from the transformation $\mathbf{f}_\theta(\mathbf{x})$ and the chosen base distribution $p_{\mathbf{z}}(\mathbf{z})$.

Proof

First we deconstruct the KL-Divergence term into its log components.

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_{\mathbf{x}}(\mathbf{x};\theta)\right] = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_{\text{data}}(\mathbf{x}) - \log p_{\mathbf{x}}(\mathbf{x};\theta)\right] \tag{C.14}$$

If we expand $p_{\mathbf{x}}(\mathbf{x};\theta)$ with the change of variables formula.

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_{\theta}(\mathbf{x})\right] = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_{\text{data}}(\mathbf{x}) - \log p_{\mathbf{z}}(\mathbf{f}_{\theta}(\mathbf{x})) - \log|\nabla_{\mathbf{x}}\mathbf{f}_{\theta}(\mathbf{x})|\right] \tag{C.15}$$

Now we do a change of variables from the data distribution $\mathbf{x}$ to the base distribution $\mathbf{z}$.

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{z})||p_{\mathbf{x}}(\mathbf{x};\theta)\right] = \mathbb{E}_{p_{\text{target}}(\mathbf{z})}\left[\log p_{\text{data}}(g_{\theta}(\mathbf{z})) - \log p_{\mathbf{z}}(\mathbf{z}) + \log|\nabla_{\mathbf{z}}g_{\theta}(\mathbf{z})|\right] \tag{C.16}$$

Recognize that we have changed the expectations from the data to the induced distribution and all terms are wrt to $\mathbf{z}$. So we can reduce this to:

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_{\mathbf{x}}(\mathbf{x};\theta)\right] = \mathbb{E}_{p_{\text{target}}(\mathbf{z})}\left[\log p_{\text{target}}(\mathbf{z}) - \log p_{\mathbf{z}}(\mathbf{z})\right] \tag{C.17}$$

where $p_{\text{target}}(\mathbf{x})$ is the distribution of $\mathbf{z} = \mathbf{f}_{\theta}(\mathbf{x})$ when $\mathbf{x}$ is sampled from $p_{\text{data}}(\mathbf{x})$. So this is simply the KL-Divergence between the transformed data in the latent space and the base distribution we choose:

$$\mathrm{D}_{\mathrm{KL}}\left[p_{\text{data}}(\mathbf{x})||p_{\theta}(\mathbf{x})\right] = \mathrm{D}_{\mathrm{KL}}\left[p_{\mathbf{f}_{\theta}}(\mathbf{z})||p_{\mathbf{z}}(\mathbf{z})\right] \tag{C.18}$$

which completes the proof.

Bibliography

Kernel methods through the roof: handling billions of points efficiently, 2020.

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In K. Keeton and T. Roscoe, editors, *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016*, pages 265–283. USENIX Association, 2016. URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. nov 2017. URL http://arxiv.org/abs/1711.06104.

M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Sy21R9JAW.

S. Andrea, B. Philip, B. W. Edward, and S. Pawel. Climate models as economic guides: Scientific challenge or quixotic quest? In *Issues in Science and Technology*, volume 31, 2015.

L. Ardizzone, R. Mackowiak, C. Rother, and U. Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/593906af0d138e69f49d251d3e7cbed0-Abstract.html.

A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 58:82–115, 2020. doi: 10.1016/j.inffus.2019.12.012. URL https://doi.org/10.1016/j.inffus.2019.12.012.

D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K. Müller. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831, 2010. URL http://portal.acm.org/citation.cfm?id=1859912.

A. G. Ballantyne. Climate change communication: what can we learn from communication theory? *WIREs Climate Change*, 7(3):329–344, 2016. doi: https://doi.org/10.1002/wcc.392. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wcc.392.

T. Baltrusaitis, C. Ahuja, and L. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):423–443, 2019. doi: 10.1109/TPAMI.2018.2798607. URL https://doi.org/10.1109/TPAMI.2018.2798607.

P. Barbiero, G. Squillero, and A. P. Tonda. Modeling generalization in machine learning: A methodological and computational study. *CoRR*, abs/2006.15680, 2020. URL https://arxiv.org/abs/2006.15680.

M. Bauer, M. van der Wilk, and C. E. Rasmussen. Understanding probabilistic sparse gaussian process approximations. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1525–1533, 2016. URL http://papers.nips.cc/paper/6477-understanding-probabilistic-sparse-gaussian-process-approximations.

M. J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University College London, UK, 2003. URL http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.404387.

J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J. Jacobsen. Invertible residual networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 573–582. PMLR, 2019. URL http://proceedings.mlr.press/v97/behrmann19a.html.

I. Belghazi, S. Rajeswar, A. Baratin, R. D. Hjelm, and A. C. Courville. MINE: mutual information neural estimation. *CoRR*, abs/1801.04062, 2018. URL http://arxiv.org/abs/1801.04062.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine learning and the bias-variance trade-off. *CoRR*, abs/1812.11118, 2018a. URL http://arxiv.org/abs/1812.11118.

M. Belkin, S. Ma, and S. Mandal. To understand deep learning we need to understand kernel learning. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 540–548. PMLR, 2018b. URL http://proceedings.mlr.press/v80/belkin18a.html.

M. Belkin, D. Hsu, and J. Xu. Two models of double descent for weak features. *SIAM J. Math. Data Sci.*, 2(4):1167–1180, 2020. doi: 10.1137/20M1336072. URL https://doi.org/10.1137/20M1336072.

K. J. Bergen, P. A. Johnson, M. V. de Hoop, and G. C. Beroza. Machine learning for data-driven discovery in solid earth geoscience. *Science*, 363(6433), 2019. ISSN 0036-8075. doi: 10.1126/science.aau0323. URL https://science.sciencemag.org/content/363/6433/eaau0323.

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Rev.*, 59(1):65–98, 2017. doi: 10.1137/141000671. URL https://doi.org/10.1137/141000671.

H. Bijl. Gaussian Process Regression Techniques. page 347, 2018. URL https://github.com/HildoBijl/GPRT.

A. Binder, G. Montavon, S. Lapuschkin, K. Müller, and W. Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In A. E. P. Villa, P. Masulli, and A. J. P. Rivero, editors, *Artificial Neural Networks and Machine Learning - ICANN 2016 - 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II*, volume 9887 of *Lecture Notes in Computer Science*, pages 63–71. Springer, 2016. doi: 10.1007/978-3-319-44781-0_8. URL https://doi.org/10.1007/978-3-319-44781-0_8.

C. M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007. ISBN 9780387310732. URL https://www.worldcat.org/oclc/71008143.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *CoRR*, abs/1601.00670, 2016. URL http://arxiv.org/abs/1601.00670.

K. Blix, G. Camps-Valls, and R. Jenssen. Gaussian process sensitivity analysis for oceanic chlorophyll estimation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10:1265–1277, 2017.

D. Blumstein, G. Chalon, T. Carlier, C. Buil, P. Hebert, T. Maciaszek, G. Ponce, T. Phulpin, B. Tournier, D. Simeoni, P. Astruc, A. Clauss, G. Kayal, and R. Jegou. IASI instrument: technical overview and measured performances. In M. Strojnik, editor, *Infrared Spaceborne Remote Sensing XII*, volume 5543, pages 196 – 207. International Society for Optics and Photonics, SPIE, 2004. doi: 10.1117/12.560907. URL https://doi.org/10.1117/12.560907.

V. I. Bogachev, A. V. Kolesnikov, and K. V. Medvedev. Triangular transformations of measures. *Sbornik: Mathematics*, 196(3):309–335, apr 2005. doi: 10.1070/sm2005v196n03abeh000882. URL https://doi.org/10.1070/sm2005v196n03abeh000882.

E. M. Bollt, J. Sun, and J. Runge. Introduction to focus issue: Causation inference and information flow in dynamical systems: Theory and applications. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075201, 2018. doi: 10.1063/1.5046848. URL https://doi.org/10.1063/1.5046848.

E. Borgonovo and E. Plischke. Sensitivity analysis: A review of recent advances, feb 2016. ISSN 03772217.

E. Borgonovo and C. L. Smith. A study of interactions in the risk assessment of complex engineering systems: An application to space psa. *Operations Research*, 59(6):1461–1476, 2011. ISSN 0030364X, 15265463. URL http://www.jstor.org/stable/41316049.

J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino. Combining data assimilation and machine learning to infer unresolved scale parametrisation, 2020.

L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

W. Bruinsma, E. Perim, W. Tebbutt, J. S. Hosking, A. Solin, and R. E. Turner. Scalable exact inference in multi-output gaussian processes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1190–1201. PMLR, 2020. URL http://proceedings.mlr.press/v119/bruinsma20a.html.

T. D. Bui, J. Yan, and R. E. Turner. A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *J. Mach. Learn. Res.*, 18:104:1–104:72, 2017a. URL http://jmlr.org/papers/v18/16-603.html.

T. D. Bui, J. Yan, and R. E. Turner. A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *J. Mach. Learn. Res.*, 18:104:1–104:72, 2017b. URL http://jmlr.org/papers/v18/16-603.html.

F. Campolongo, J. Cariboni, and A. Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental Modelling and Software*, 22(10):1509–1518, 2007. ISSN 13648152. doi: 10.1016/j.envsoft.2006.10.004.

G. Camps-Valls, L. Gómez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla. Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 3:93–97, 2006.

G. Camps-Valls, J. Muñoz and, L. Gómez-Chova, L. Guanter, and X. Calbet. Nonlinear statistical retrieval of atmospheric profiles from MetOp-IASI and MTG-IRS infrared sounding data. *IEEE Trans. Geosci. Rem. Sens.*, 50(5):1759–1769, 2012.

G. Camps-Valls, J. Verrelst, J. Munoz-Mari, V. Laparra, F. Mateo-Jimenez, and J. Gomez-Dans. A survey on gaussian processes for earth-observation data analysis: A comprehensive investigation. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):58–78, 2016. doi: 10.1109/MGRS.2015.2510084.

J. Q. Candela. *Learning with uncertainty-Gaussian processes and relevance vector machines*. PhD thesis, 2004. URL http://eprints.pascal-network.org/archive/00000896/.

J. Q. Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959, 2005. URL http://jmlr.org/papers/v6/quinonero-candela05a.html.

N. D. Cao, W. Aziz, and I. Titov. Block neural autoregressive flow. In A. Globerson and R. Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 1263–1273. AUAI Press, 2019. URL http://proceedings.mlr.press/v115/de-cao20a.html.

J.-F. Cardoso. Dependence, Correlation and Gaussianity in Independent Component Analysis. *Journal of Machine Learning Research*, 4:1177–1203, 2003.

G. Chalon, F. Cayla, and D. Diebel. IASI: an advanced sounder for operational meteorology. In *Proceedings of the 52nd Congress of IAF*, Toulouse, France, 2001.

K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1906995116. URL https://www.pnas.org/content/116/45/22445.

C. Chan, A. Al-Bashabsheh, H. P. Huang, M. Lim, D. S. H. Tam, and C. Zhao. Neural entropic estimation: A faster path to mutual information estimation. *CoRR*, abs/1905.12957, 2019. URL http://arxiv.org/abs/1905.12957.

B. Charlier, J. Feydy, J. A. Glaunès, F.-D. Collin, and G. Durif. Kernel operations on the GPU, with autodiff, without memory overflows. *arXiv preprint arXiv:2004.11127*, 2020.

S. S. Chen and R. A. Gopinath. Gaussianization. In *Advances in Neural Information Processing Systems*, 2001. ISBN 0262122413. doi: 10.1007/978-88-470-2706-0_30.

T. Chen, E. B. Fox, and C. Guestrin. Stochastic gradient hamiltonian monte carlo. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1683–1691. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/cheni14.html.

T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, abs/1512.01274, 2015. URL http://arxiv.org/abs/1512.01274.

T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6572–6583, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

T. Q. Chen, J. Behrmann, D. Duvenaud, and J. Jacobsen. Residual flows for invertible generative modeling. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9913–9923, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5d0d5594d24f0f955548f0fc0ff83d10-Abstract.html.

Z. Chen and B. Wang. How priors of initial hyperparameters affect gaussian process regression models. *Neurocomputing*, 275:1702–1710, 2018. doi: 10.1016/j.neucom.2017.10.028. URL https://doi.org/10.1016/j.neucom.2017.10.028.

C. Cheng and B. Kingsbury. Arccosine kernels: Acoustic modeling with infinite neural networks. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5200–5203, 2011. doi: 10.1109/ICASSP.2011.5947529.

S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995. ISSN 00031305. URL http://www.jstor.org/stable/2684568.

D. Colquhoun. An investigation of the false discovery rate and the misinterpretation of p-values. *Royal Society Open Science*, 1(3):140216, 2014. doi: 10.1098/rsos.140216. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsos.140216.

C. D. Correa and P. Lindstrom. The mutual information diagram for uncertainty visualization. *International Journal for Uncertainty Quantification*, 3:187–201, 2013a.

C. D. Correa and P. Lindstrom. The mutual information diagram for uncertainty visualization. *International Journal for Uncertainty Quantification*, 3:187–201, 2013b.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2001. ISBN 9780471062592. doi: 10.1002/0471200611. URL https://doi.org/10.1002/0471200611.

M. D. Cranmer, A. Sanchez-Gonzalez, P. W. Battaglia, R. Xu, K. Cranmer, D. N. Spergel, and S. Ho. Discovering symbolic models from deep learning with inductive biases. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c9f2f917078bd2db12f23c3b413d9cba-Abstract.html.

A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.*, 35(1):53–65, 2018. doi: 10.1109/MSP.2017.2765202. URL https://doi.org/10.1109/MSP.2017.2765202.

K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone. Random feature expansions for deep gaussian processes. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 884–893. PMLR, 2017. URL http://proceedings.mlr.press/v70/cutajar17a.html.

S. Da Veiga. Global sensitivity analysis with dependence measures. *Journal of Statistical Computation and Simulation*, 85(7):1283–1305, may 2015. ISSN 0094-9655. doi: 10.1080/00949655.2014.945932. URL https://hal.archives-ouvertes.fr/hal-00903283http://www.tandfonline.com/doi/abs/10.1080/00949655.2014.945932.

B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M. Balcan, and L. Song. Scalable kernel methods via doubly stochastic gradients. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3041–3049, 2014. URL http://papers.nips.cc/paper/5238-scalable-kernel-methods-via-doubly-stochastic-gradients.

P. Dallaire, C. Besse, and B. Chaib-draa. An approximate inference with Gaussian process to latent functions from uncertain data. *Neurocomputing*, 74(11):1945–1955, 2011. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2010.09.024. URL http://www.sciencedirect.com/science/article/pii/S0925231211000440.

A. Damianou. *Deep Gaussian Processes and Variational Propagation of Uncertainty*. PhD thesis, 2015.

A. C. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Inference for Uncertainty on the Inputs of Gaussian Process Models. pages 1–51, 2014. URL http://arxiv.org/abs/1409.2287.

A. C. Damianou, M. K. Titsias, and N. D. Lawrence. Variational inference for latent variables and uncertain inputs in gaussian processes. *J. Mach. Learn. Res.*, 17:42:1–42:62, 2016. URL http://jmlr.org/papers/v17/damianou16a.html.

A. DÁmour, K. A. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, F. Hormozdiari, N. Houlsby, S. Hou, G. Jerfel, A. Karthikesalingam, M. Lucic, Y. Ma, C. McLean, D. Mincu, A. Mitani, A. Montanari, Z. Nado, V. Natarajan, C. Nielson, T. F. Osborne, R. Raman, K. Ramasamy, R. Sayres, J. Schrouff, M. Seneviratne, S. Sequeira, H. Suresh, V. Veitch, M. Vladymyrov, X. Wang, K. Webster, S. Yadlowsky, T. Yun, X. Zhai, and D. Sculley. Underspecification presents

challenges for credibility in modern machine learning. *CoRR*, abs/2011.03395, 2020. URL https://arxiv.org/abs/2011.03395.

A. G. de G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018a. URL https://openreview.net/forum?id=H1-nGgWC-.

A. G. de G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018b. URL https://openreview.net/forum?id=H1-nGgWC-.

D. A. R. M. A. de Souza, C. L. C. Mattos, and J. P. P. Gomes. Unscented gaussian process latent variable model: learning from uncertain inputs with intractable kernels. *CoRR*, abs/1907.01867, 2019. URL http://arxiv.org/abs/1907.01867.

D. A. R. M. A. de Souza, D. Mesquita, C. L. C. Mattos, and J. P. P. Gomes. Learning gplvm with arbitrary kernels using the unscented transformation, 2020.

M. Deisenroth. *Efficient Reinforcement Learning using Gaussian Processes*. PhD thesis, 2010.

M. P. Deisenroth and S. Mohamed. Expectation Propagation in Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems*, volume 4, pages 2609–2617, 2012. ISBN 9781627480031. URL http://arxiv.org/abs/1207.2940.

P. Dellaportas and D. A. Stephens. Bayesian Analysis of Errors-in-Variables Regression Models. *Biometrics*, 51(3):1085, sep 1995. ISSN 0006341X. doi: 10.2307/2533007. URL https://www.jstor.org/stable/2533007?origin=crossref.

B. Dennis, J. M. Ponciano, M. L. Taper, and S. R. Lele. Errors in statistical inference under model misspecification: Evidence, hypothesis testing, and aic. *Frontiers in Ecology and Evolution*, 7:372, 2019. ISSN 2296-701X. doi: 10.3389/fevo.2019.00372. URL https://www.frontiersin.org/article/10.3389/fevo.2019.00372.

L. Dinh, D. Krueger, and Y. Bengio. NICE: non-linear independent components estimation. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. URL http://arxiv.org/abs/1410.8516.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016. URL http://arxiv.org/abs/1605.08803.

H. M. Dolatabadi, S. M. Erfani, and C. Leckie. Invertible generative modeling using linear rational splines. In S. Chiappa and R. Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 4236–4246. PMLR, 2020. URL http://proceedings.mlr.press/v108/dolatabadi20a.html.

D. Douglas-Smith, T. Iwanaga, B. F. W. Croke, and A. J. Jakeman. Certain trends in uncertainty and sensitivity analysis: An overview of software tools and techniques. *Environmental Modelling and Software*, 124:104588, 2020. doi: 10.5281/zenodo.3406946. URL http://creativecommons.org/licenses/by/4.0/.

C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7509–7520, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/7ac71d433f282034e088473244df8c02-Abstract.html.

C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. nflows: normalizing flows in PyTorch, Nov. 2020. URL https://doi.org/10.5281/zenodo.4296287.

V. Dutordoir. Processes Non-Stationary Surrogate Modeling with Deep Gaussian. 2016.

V. Dutordoir, H. Salimbeni, M. P. Deisenroth, and J. Hensman. Gaussian process conditional density estimation. In *Advances in Neural Information Processing Systems*, volume 2018-Decem, pages 2385–2395, 2018. URL http://arxiv.org/abs/1810.12750.

G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In M. Meila and T. Heskes, editors, *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, pages 258–267. AUAI Press, 2015. URL http://auai.org/uai2015/proceedings/papers/230.pdf.

N. Endou. Fubini's theorem. *Formaliz. Math.*, 27(1):67–74, 2019. doi: 10.2478/forma-2019-0007. URL https://doi.org/10.2478/forma-2019-0007.

V. Eyring, M. Righi, A. Lauer, M. Evaldsson, S. Wenzel, C. Jones, A. Anav, O. Andrews, I. Cionni, E. L. Davin, C. Deser, C. Ehbrecht, P. Friedlingstein, P. Gleckler, K.-D. Gottschaldt, S. Hagemann, M. Juckes, S. Kindermann, J. Krasting, D. Kunert, R. Levine, A. Loew, J. Mäkelä, G. Martin, E. Mason, A. S. Phillips, S. Read, C. Rio, R. Roehrig, D. Senftleben, A. Sterl, L. H. van Ulft, J. Walton, S. Wang, and K. D. Williams. Esmvaltool (v1.0) – a community diagnostic and performance metrics tool for routine evaluation of earth system models in cmip. *Geoscientific Model Development*, 9(5):1747–1802, 2016. doi: 10.5194/gmd-9-1747-2016. URL https://gmd.copernicus.org/articles/9/1747/2016/.

L. S. Freedman, D. Midthune, R. J. Carroll, and V. Kipnis. A comparison of regression calibration, moment reconstruction and imputation for adjusting for covariate measurement error in regression. *Statistics in Medicine*, 27(25):5195–5216, nov 2008. ISSN 02776715. doi: 10.1002/sim.3361. URL http://doi.wiley.com/10.1002/sim.3361.

M. L. Freedman, D. Reich, K. L. Penney, G. J. McDonald, A. A. Mignault, N. Patterson, S. B. Gabriel, E. J. Topol, J. W. Smoller, C. N. Pato, M. T. Pato, T. L. Petryshen, L. N. Kolonel, E. S. Lander, P. Sklar, B. Henderson, J. N. Hirschhorn, and D. Altshuler. Assessing the impact of population stratification on genetic association studies. *Nature Genetics*, 36(4):388–393, 2004. ISSN 10614036. doi: 10.1038/ng1333.

K. J. Friston, J. Mattout, N. J. Trujillo-Barreto, J. Ashburner, and W. D. Penny. Variational free energy and the laplace approximation. *NeuroImage*, 34(1):220–234, 2007. doi: 10.1016/j.neuroimage.2006.08.035. URL https://doi.org/10.1016/j.neuroimage.2006.08.035.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In M. Balcan and K. Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/gal16.html.

J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/27e8e17134dd7083b050476733207ea1-Paper.pdf.

A. Gelman, A. Vehtari, D. Simpson, C. C. Margossian, B. Carpenter, Y. Yao, L. Kennedy, J. Gabry, P.-C. Bürkner, and M. Modrák. Bayesian workflow, 2020.

Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nat.*, 521 (7553):452–459, 2015. doi: 10.1038/nature14541. URL https://doi.org/10.1038/nature14541.

A. Girard. Approximate methods for propagation of uncertainty with Gaussian process models. *Ph.D. Thesis*, (October), 2004. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.66.1826&rep=rep1&type=pdf.

A. Girard and R. Murray-Smith. Learning a Gaussian Process Model with Uncertain Inputs. (c):1–10, 2003. URL http://www.dcs.gla.ac.uk/$\sim$rod/publications/GirMur03-tr-144.pdf.

A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs - application to multiple-step ahead

time series forecasting. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 529–536. MIT Press, 2002a. URL http://papers.nips.cc/paper/2313-gaussian-process-priors-with-uncertain-inputs-application-to-multiple-step-ahead-tim

A. Girard, C. E. Rasmussen, and R. Murray-Smith. Gaussian Process priors with Uncertain Inputs : Multiple-Step-Ahead Prediction. *Technical Report TR-2002-119*, pages 1–18, 2002b.

M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *J. Mach. Learn. Res.*, 12: 2211–2268, 2011. URL http://dl.acm.org/citation.cfm?id=2021071.

W. Gong, H. Gupta, D. Yang, K. Sricharan, and A. Hero. Estimating epistemic and aleatory uncertainties during hydrologic modeling: An information theoretic approach. *Water Resources Research*, 49:2253–2273, 2013.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014. URL http://arxiv.org/abs/1406.2661.

W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. FFJORD: free-form continuous dynamics for scalable reversible generative models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJxgknCcK7.

A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 6645–6649. IEEE, 2013. doi: 10.1109/ICASSP.2013.6638947. URL https://doi.org/10.1109/ICASSP.2013.6638947.

A. Gretton, R. Herbrich, A. J. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *J. Mach. Learn. Res.*, 6:2075–2129, 2005. URL http://jmlr.org/papers/v6/gretton05a.html.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012. URL http://dl.acm.org/citation.cfm?id=2188410.

J. H. A. Guillaume, J. D. Jakeman, S. Marsili-Libelli, M. Asher, P. Brunner, B. F. W. Croke, M. C. Hill, A. J. Jakeman, K. J. Keesman, S. Razavi, and J. D. Stigter. Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose. *Environ. Model. Softw.*, 119:418–432, 2019. doi: 10.1016/j.envsoft.2019.07.007. URL https://doi.org/10.1016/j.envsoft.2019.07.007.

C. Guo, J. S. Frank, and K. Q. Weinberger. Low frequency adversarial perturbation. In A. Globerson and R. Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 1127–1137. AUAI Press, 2019. URL http://proceedings.mlr.press/v115/guo20a.html.

H. Gupta and S. Razavi. Revisiting the basis of sensitivity analysis for dynamical earth system models. *Water Resources Research*, 54:8692–8717, 2018.

N. Guttman. Statistical descriptors of climate. *Bulletin of the American Meteorological Society*, 70:602–607, 1989.

D. Hafner, D. Tran, T. P. Lillicrap, A. Irpan, and J. Davidson. Noise contrastive priors for functional uncertainty. In A. Globerson and R. Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 905–914. AUAI Press, 2019. URL http://proceedings.mlr.press/v115/hafner20a.html.

J. W. Hardin, H. Schmeidiche, and R. J. Carroll. The regression-calibration method for fitting generalized linear models with additive measurement error. *Stata Journal*, 3(4):373–385, December 2003. URL https://ideas.repec.org/a/tsj/stataj/v3y2003i4p373-385.html.

M. Havasi, J. M. Hernández-Lobato, and J. J. Murillo-Fuentes. Inference in deep gaussian processes using stochastic gradient hamiltonian monte carlo. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7517–7527, 2018. URL http://papers.nips.cc/paper/7979-inference-in-deep-gaussian-processes-using-stochastic-gradient-hamiltonian-monte-car

B. He, B. Lakshminarayanan, and Y. W. Teh. Bayesian deep ensembles via the neural tangent kernel. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0b1ec366924b26fc98fa7b71a9c249cf-Abstract.html.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.123. URL https://doi.org/10.1109/ICCV.2015.123.

J. C. Helton and F. J. Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliab. Eng. Syst. Saf.*, 81(1):23–69, 2003. doi: 10.1016/S0951-8320(03)00058-9. URL https://doi.org/10.1016/S0951-8320(03)00058-9.

J. Hensman, A. G. De Matthews, M. Filippone, and Z. Ghahramani. MCMC for variationally sparse Gaussian processes. *Advances in Neural Information Processing Systems*, 2015-Janua:1648–1656, 2015. ISSN 10495258.

F. Heße, A. Comunian, and S. Attinger. What we talk about when we talk about uncertainty. toward a unified, data-driven framework for uncertainty characterization in hydrogeology. *Frontiers in Earth Science*, 7:118, 2019.

F. Hilton, R. Armante, T. August, C. Barnet, A. Bouchard, C. Camy-Peyret, V. Capelle, L. Clarisse, C. Clerbaux, P.-F. Coheur, A. Collard, C. Crevoisier, G. Dufour, D. Edwards, F. Faijan, N. Fourri?, A. Gambacorta, M. Goldberg, V. Guidard, D. Hurtmans, S. Illingworth, N. Jacquinet-Husson, T. Kerzenmacher, D. Klaes, L. Lavanant, G. Masiello, M. Matricardi, A. McNally, S. Newman, E. Pavelin, S. Payan, E. P?quignot, S. Peyridieu, T. Phulpin, J. Remedios, P. Schl?ssel, C. Serio, L. Strow, C. Stubenrauch, J. Taylor, D. Tobin, W. Wolf, and D. Zhou. Hyperspectral earth observation from iasi: Five years of accomplishments. *Bulletin of the American Meteorological Society*, 93(3):347 – 370, 01 Mar. 2012. doi: 10.1175/BAMS-D-11-00027.1. URL https://journals.ametsoc.org/view/journals/bams/93/3/bams-d-11-00027_1.xml.

J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2722–2730. PMLR, 2019. URL http://proceedings.mlr.press/v97/ho19a.html.

M. Hoffman, P. Sountsov, J. V. Dillon, I. Langmore, D. Tran, and S. Vasudevan. Neutralizing bad geometry in hamiltonian monte carlo using neural transport. *arXiv: Computation*, 2019.

M. D. Hoffman and A. Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *CoRR*, abs/1111.4246, 2011. URL http://arxiv.org/abs/1111.4246.

M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014. URL http://dl.acm.org/citation.cfm?id=2638586.

E. Hoogeboom, R. van den Berg, and M. Welling. Emerging convolutions for generative normalizing flows. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach,*

California, USA, volume 97 of *Proceedings of Machine Learning Research*, pages 2771–2780. PMLR, 2019. URL http://proceedings.mlr.press/v97/hoogeboom19a.html.

J. Hron, Y. Bahri, J. Sohl-Dickstein, and R. Novak. Infinite attention: NNGP and NTK for deep attention networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4376–4386. PMLR, 2020. URL http://proceedings.mlr.press/v119/hron20a.html.

C. Huang, D. Krueger, A. Lacoste, and A. C. Courville. Neural autoregressive flows. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2083–2092. PMLR, 2018. URL http://proceedings.mlr.press/v80/huang18d.html.

P. Huang, H. Avron, T. N. Sainath, V. Sindhwani, and B. Ramabhadran. Kernel methods match deep neural networks on TIMIT. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, pages 205–209. IEEE, 2014. doi: 10.1109/ICASSP.2014.6853587. URL https://doi.org/10.1109/ICASSP.2014.6853587.

D. Huard and A. Mailhot. A bayesian perspective on input uncertainty in model calibration: Application to hydrological model "abc". *Water Resources Research*, 42(7), 2006. doi: https://doi.org/10.1029/2005WR004661. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005WR004661.

I. L. Hudson. Data integration using advances in machine learning in drug discovery and molecular biology. In H. M. Cartwright, editor, *Artificial Neural Networks - Third Edition*, volume 2190 of *Methods in Molecular Biology*, pages 167–184. Springer, 2021. doi: 10.1007/978-1-0716-0826-5_7. URL https://doi.org/10.1007/978-1-0716-0826-5_7.

F. Huszar. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *CoRR*, abs/1511.05101, 2015. URL http://arxiv.org/abs/1511.05101.

D. I. Inouye and P. Ravikumar. Deep density destructors. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2172–2180. PMLR, 2018. URL http://proceedings.mlr.press/v80/inouye18a.html.

S. Intergovernmental Panel on Climate Change. Working Group 1, M. Susan Solomon, S. Solomon, G. d'experts intergouvernemental sur l'évolution du climat, I. P. on Climate Change, I. P. on Climate Change. Working Group Science, W. Change, D. Qin, S. Salomon, M. Manning, et al. *Climate Change 2007 - The Physical Science Basis: Working Group I Contribution to the Fourth Assessment Report of the IPCC*. Assessment re-

port (Intergovernmental Panel on Climate Change).: Working Group. Cambridge University Press, 2007. ISBN 9780521705967. URL https://books.google.es/books?id=8-m8nXB8GB4C.

T. Iwanaga, D. Partington, J. Ticehurst, B. F. Croke, and A. J. Jakeman. A socio-environmental model for exploring sustainable water management futures: Participatory and collaborative modelling in the lower campaspe catchment. *Journal of Hydrology: Regional Studies*, 28:100669, 2020. ISSN 2214-5818. doi: https://doi.org/10.1016/j.ejrh.2020.100669. URL http://www.sciencedirect.com/science/article/pii/S2214581819303726.

P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. In A. Globerson and R. Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 876–885. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/313.pdf.

P. Izmailov, W. Maddox, P. Kirichenko, T. Garipov, D. P. Vetrov, and A. G. Wilson. Subspace inference for bayesian deep learning. In A. Globerson and R. Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 1169–1179. AUAI Press, 2019. URL http://proceedings.mlr.press/v115/izmailov20a.html.

E. Izquierdo-Verdiguier, R. Jenssen, L. Gómez-Chova, and G. Camps-Valls. Spectral clustering with the probabilistic cluster kernel. *Neurocomputing*, 149:1299–1304, 2015. doi: 10.1016/j.neucom.2014.08.068. URL https://doi.org/10.1016/j.neucom.2014.08.068.

E. Izquierdo-Verdiguier, V. Laparra, R. Jenssen, L. Gómez-Chova, and G. Camps-Valls. Optimized kernel entropy components. *IEEE transactions on neural networks and learning systems*, 28 6:1466–1472, 2017.

A. Jacot, C. Hongler, and F. Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8580–8589, 2018. URL http://papers.nips.cc/paper/8076-neural-tangent-kernel-convergence-and-generalization-in-neural-networks.

P. Jaini, I. Kobyzev, Y. Yu, and M. Brubaker. Tails of lipschitz triangular flows. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4673–4681. PMLR, 2020. URL http://proceedings.mlr.press/v119/jaini20a.html.

M. Jankowiak, G. Pleiss, and J. R. Gardner. Deep sigma point processes. In R. P. Adams and V. Gogate, editors, *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI 2020, virtual online, August 3-6, 2020*, volume 124 of *Proceedings of Machine Learning Research*, pages 789–798. AUAI Press, 2020. URL http://proceedings.mlr.press/v124/jankowiak20a.html.

R. Jenssen. Kernel entropy component analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):847–860, 2010. doi: 10.1109/TPAMI.2009.100. URL https://doi.org/10.1109/TPAMI.2009.100.

X. Jia, J. Willard, A. Karpatne, J. Read, J. Zwart, M. Steinbach, and V. Kumar. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ArXiv*, abs/2001.11086, 2020.

J. Jiménez-Luna, F. Grisoni, and G. Schneider. Drug discovery with explainable artificial intelligence. *CoRR*, abs/2007.00523, 2020. URL https://arxiv.org/abs/2007.00523.

J. E. Johnson, V. Laparra, and G. Camps-Valls. Accounting for input noise in gaussian process parameter retrieval. *IEEE Geosci. Remote. Sens. Lett.*, 17(3):391–395, 2020a. doi: 10.1109/LGRS.2019.2921476. URL https://doi.org/10.1109/LGRS.2019.2921476.

J. E. Johnson, V. Laparra, G. Camps-Valls, R. Santos-Rodríguez, and J. Malo. Information Theory in Density Destructors. (3), 2020b. URL http://isp.uv.es/rbig.htmlhttp://arxiv.org/abs/2012.01012.

J. E. Johnson, V. Laparra, M. Piles, and G. Camps-Valls. Gaussianizing the Earth: Multidimensional Information Measures for Earth Data Analysis. 2020c. doi: 10.1109/MGRS.2021.3066260. URL http://arxiv.org/abs/2010.06476.

J. E. Johnson, V. Laparra, A. Pérez-Suay, M. D. Mahecha, and G. Camps-Valls. Kernel methods and their derivatives: Concept and perspectives for the earth system sciences. *PLOS ONE*, 15(10):1–30, 10 2020d. doi: 10.1371/journal.pone.0235885. URL https://doi.org/10.1371/journal.pone.0235885.

P. Jylänki, J. Vanhatalo, and A. Vehtari. Robust gaussian process regression with a student-t likelihood. *J. Mach. Learn. Res.*, 12:3227–3257, 2011. URL http://dl.acm.org/citation.cfm?id=2078209.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *CoRR*, abs/1807.02582, 2018a. URL http://arxiv.org/abs/1807.02582.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *CoRR*, abs/1807.02582, 2018b. URL http://arxiv.org/abs/1807.02582.

M. G. Kapteyn, D. J. Knezevic, and K. Willcox. *Toward predictive digital twins via component-based reduced-order models and interpretable machine learning*. doi: 10.2514/6.2020-0418. URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-0418.

T. Karaletsos and T. D. Bui. Hierarchical gaussian process priors for bayesian neural network weights. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c70341de2c112a6b3496aec1f631dddd-Abstract.html.

K. Kawaguchi, L. P. Kaelbling, and Y. Bengio. Generalization in deep learning. *CoRR*, abs/1710.05468, 2017. URL http://arxiv.org/abs/1710.05468.

M. Kendall and A. Stuart. *The Advanced Theory of Statistics*. Number v. 2 in The Advanced Theory of Statistics. Charles Griffin, 1963. ISBN 9780852640722. URL https://books.google.es/books?id=ARrvAAAAMAAJ.

K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard. Most likely heteroscedastic gaussian process regression. In Z. Ghahramani, editor, *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 393–400. ACM, 2007. doi: 10.1145/1273496.1273546. URL https://doi.org/10.1145/1273496.1273546.

D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10236–10245, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/d139db6a236200b21cc7f752979132d0-Abstract.html.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Found. Trends Mach. Learn.*, 12(4):307–392, 2019. doi: 10.1561/2200000056. URL https://doi.org/10.1561/2200000056.

D. P. Kingma, T. Salimans, and M. Welling. Improving variational inference with inverse autoregressive flow. *CoRR*, abs/1606.04934, 2016. URL http://arxiv.org/abs/1606.04934.

K. H. Knuth, A. Gotera, C. T. Curry, K. A. Huyser, K. R. Wheeler, and W. B. Rossow. Revealing relationships among relevant climate variables with information theory, 2013.

I. Kobyzev, S. Prince, and M. A. Brubaker. Normalizing flows: Introduction and ideas. *CoRR*, abs/1908.09257, 2019. URL http://arxiv.org/abs/1908.09257.

I. Kobyzev, S. J. Prince, B. A. Marcus, and M. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. ISSN 0162-8828. doi: 10.1109/tpami.2020.2992934.

R. E. Kopp, A. C. Kemp, K. Bittermann, B. P. Horton, J. P. Donnelly, W. R. Gehrels, C. C. Hay, J. X. Mitrovica, E. D. Morrow, and S. Rahmstorf. Temperature-driven global sea-level variability in the common era. *Proceedings of the National Academy of Sciences*, 113 (11):E1434–E1441, 2016. ISSN 0027-8424. doi: 10.1073/pnas.1517056113. URL https://www.pnas.org/content/113/11/E1434.

S. Kornblith, M. Norouzi, H. Lee, and G. E. Hinton. Similarity of neural network representations revisited. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 2019. URL http://proceedings.mlr.press/v97/kornblith19a.html.

K. Krauth, E. V. Bonilla, K. Cutajar, and M. Filippone. Autogp: Exploring the capabilities and limitations of gaussian process models. In G. Elidan, K. Kersting, and A. T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017. URL http://auai.org/uai2017/proceedings/papers/50.pdf.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017. doi: 10.1145/3065386. URL http://doi.acm.org/10.1145/3065386.

S. Kucherenko and B. Iooss. Derivative based global sensitivity measures. dec 2014. URL http://arxiv.org/abs/1412.2619.

S. Kucherenko and S. Song. Derivative-based global sensitivity measures and their link with sobol' sensitivity indices. In *Springer Proceedings in Mathematics and Statistics*, volume 163, pages 455–469, may 2016. ISBN 9783319335056. doi: 10.1007/978-3-319-33507-0_23. URL http://arxiv.org/abs/1605.07830http://dx.doi.org/10.1007/978-3-319-33507-0_23.

P. Kumar and H. V. Gupta. Debates—does information theory provide a new paradigm for earth science? *Water Resources Research*, 56(2):e2019WR026398, 2020. doi: https://doi.org/10.1029/2019WR026398. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR026398. e2019WR026398 2019WR026398.

A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HJGU3Rodl.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6402–6413, 2017. URL http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.

V. Lalchand, A. Ravuri, and N. D. Lawrence. Gaussian process latent variable flows for massively missing data. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021. URL https://openreview.net/forum?id=zaMwvOjsyym.

J. Lao, C. Suter, I. Langmore, C. Chimisov, A. Saxena, P. Sountsov, D. Moore, R. A. Saurous, M. D. Hoffman, and J. V. Dillon. tfp.mcmc: Modern markov chain monte carlo tools built for modern hardware. *CoRR*, abs/2002.01184, 2020. URL https://arxiv.org/abs/2002.01184.

V. Laparra, G. Camps-Valls, and J. Malo. Iterative gaussianization: From ICA to random rotations. *IEEE Transactions on Neural Networks*, 22(4):537–549, 2011a. ISSN 10459227. doi: 10.1109/TNN.2011.2106511.

V. Laparra, G. Camps-Valls, and J. Malo. Iterative gaussianization: From ICA to random rotations. *IEEE Trans. Neural Networks*, 22(4):537–549, 2011b. doi: 10.1109/TNN.2011.2106511. URL https://doi.org/10.1109/TNN.2011.2106511.

V. Laparra, J. E. Johnson, G. Camps-Valls, R. Santos-Rodríguez, and J. Malo. Information Theory Measures via Multidimensional Gaussianization. oct 2020. URL http://arxiv.org/abs/2010.03807.

J. W. Larson. Visualizing climate variability with time-dependent probability density functions, detecting it using information theory. *Procedia Computer Science*, 9:917 – 926, 2012. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2012.04.098. URL http://www.sciencedirect.com/science/article/pii/S1877050912002190. Proceedings of the International Conference on Computational Science, ICCS 2012.

I. Lauriola and F. Aiolli. Mklpy: a python-based framework for multiple kernel learning. *CoRR*, abs/2007.09982, 2020. URL https://arxiv.org/abs/2007.09982.

N. D. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *J. Mach. Learn. Res.*, 6:1783–1816, 2005. URL http://jmlr.org/papers/v6/lawrence05a.html.

M. Lázaro-Gredilla and M. K. Titsias. Variational heteroscedastic gaussian process regression. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 841–848. Omnipress, 2011. URL https://icml.cc/2011/papers/456_icmlpaper.pdf.

Q. V. Le, T. Sarlós, and A. J. Smola. Fastfood - computing hilbert space expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 244–252. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/le13.html.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015a. doi: 10.1038/nature14539. URL https://doi.org/10.1038/nature14539.

Y. LeCun, Y. Bengio, and G. E. Hinton. Deep learning. *Nat.*, 521(7553):436–444, 2015b. doi: 10.1038/nature14539. URL https://doi.org/10.1038/nature14539.

J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as gaussian processes. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=B1EA-M-0Z.

C. Lemieux. Springer, 2009. doi: 10.1007/978-0-387-78165-5.

Y. Li and Y. Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8168–8177, 2018. URL http://papers.nips.cc/paper/8038-learning-overparameterized-neural-networks-via-stochastic-gradient-descent-on-structu

J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/543e83748234f7cbab21aa0ade66565f-Abstract.html.

S. Løkse, F. M. Bianchi, A. Salberg, and R. Jenssen. Spectral clustering using PCKID - A probabilistic cluster kernel for incomplete data. In P. Sharma and F. M. Bianchi, editors, *Image Analysis - 20th Scandinavian Conference, SCIA 2017, Tromsø, Norway, June 12-14, 2017, Proceedings, Part I*, volume 10269 of *Lecture Notes in Computer Science*, pages 431–442. Springer, 2017. doi: 10.1007/978-3-319-59126-1_36. URL https://doi.org/10.1007/978-3-319-59126-1_36.

D. P. Loucks and E. van Beek. *System Sensitivity and Uncertainty Analysis*, pages 331–374. Springer International Publishing, Cham, 2017. ISBN 978-3-319-44234-1. doi: 10.1007/978-3-319-44234-1_8. URL https://doi.org/10.1007/978-3-319-44234-1_8.

X. Lu, A. Rudi, E. Borgonovo, and L. Rosasco. Faster kriging: Facing high-dimensional simulators. *Oper. Res.*, 68(1):233–249, Jan. 2020. ISSN 0030-364X. doi: 10.1287/opre.2019.1860. URL https://doi.org/10.1287/opre.2019.1860.

Z. Lu, A. May, K. Liu, A. B. Garakani, D. Guo, A. Bellet, L. Fan, M. Collins, B. Kingsbury, M. Picheny, and F. Sha. How to scale up kernel methods to be as good as deep neural nets. *CoRR*, abs/1411.4000, 2014. URL http://arxiv.org/abs/1411.4000.

S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774, 2017a. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017b. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. Explainable ai for trees: From local explanations to global understanding. *ArXiv*, abs/1905.04610, 2019.

K. W. Ma, J. P. Lewis, and W. B. Kleijn. The HSIC bottleneck: Deep learning without back-propagation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5085–5092. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/5950.

S. Ma and M. Belkin. Diving into the shallows: a computational perspective on large-scale shallow learning. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3778–3787, 2017. URL http://papers.nips.cc/paper/6968-diving-into-the-shallows-a-computational-perspective-on-large-scale-shallow-learning

D. J. C. Mackay. Probable networks and plausible predictions — a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995. doi: 10.1088/0954-898X_6_3_011. URL https://doi.org/10.1088/0954-898X_6_3_011.

D. J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003. ISBN 978-0-521-64298-9.

W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13132–13143, 2019a. URL http://papers.nips.cc/paper/9472-a-simple-baseline-for-bayesian-uncertainty-in-deep-learning.

W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13132–13143, 2019b. URL http://papers.nips.cc/paper/9472-a-simple-baseline-for-bayesian-uncertainty-in-deep-learning.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

M. D. Mahecha, F. Gans, G. Brandt, R. Christiansen, S. E. Cornell, N. Fomferra, G. Kraemer, J. Peters, P. Bodesheim, G. Camps-Valls, J. F. Donges, W. Dorigo, L. M. Estupinan-Suarez, V. H. Gutierrez-Velez, M. Gutwin, M. Jung, M. C. Londoño, D. G. Miralles, P. Papastefanou, and M. Reichstein. Earth system data cubes unravel global multivariate dynamics. *Earth System Dynamics*, 11(1):201–234, 2020. doi: 10.5194/esd-11-201-2020. URL https://esd.copernicus.org/articles/11/201/2020/.

A. J. Majda and B. Gershgorin. Quantifying uncertainty in climate change science through empirical information theory. *Proceedings of the National Academy of Sciences*, 107(34): 14958–14963, 2010. ISSN 0027-8424. doi: 10.1073/pnas.1007009107. URL https://www.pnas.org/content/107/34/14958.

J. Maroñas, O. Hamelijnck, J. Knoblauch, and T. Damoulas. Transforming gaussian processes with normalizing flows. *CoRR*, abs/2011.01596, 2020. URL https://arxiv.org/abs/2011.01596.

M. Marzjarani. Simulation and the monte carlo method (3rd ed.). *Technometrics*, 61(3): 427–428, 2019. doi: 10.1080/00401706.2019.1629745. URL https://doi.org/10.1080/00401706.2019.1629745.

A. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. Gpflow: A gaussian process library using tensorflow. *J. Mach. Learn. Res.*, 18:40:1–40:6, 2017. URL http://jmlr.org/papers/v18/16-537.html.

A. May, A. B. Garakani, Z. Lu, D. Guo, K. Liu, A. Bellet, L. Fan, M. Collins, D. Hsu, B. Kingsbury, M. Picheny, and F. Sha. Kernel approximation methods for speech recognition. *J. Mach. Learn. Res.*, 20:59:1–59:36, 2019. URL http://jmlr.org/papers/v20/17-026.html.

A. McHutchon. Nonlinear Modelling and Control using Gaussian Processes. *Thesis*, (August), 2014. URL .http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.703.4461%0Ahttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.703.4461%0A.

A. McHutchon and C. E. Rasmussen. Gaussian process training with input noise. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1341–1349, 2011. URL http://papers.nips.cc/paper/4295-gaussian-process-training-with-input-noise.

S. A. McQuarrie, C. Huang, and K. Willcox. Data-driven reduced-order models via regularized operator inference for a single-injector combustion process. *CoRR*, abs/2008.02862, 2020. URL https://arxiv.org/abs/2008.02862.

G. Meanti, L. Carratino, L. Rosasco, and A. Rudi. Kernel methods through the roof: Handling billions of points efficiently. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a59afb1b7d82ec353921a55c579ee26d-Abstract.html.

P. A. Mendoza, M. P. Clark, M. Barlage, B. Rajagopalan, L. Samaniego, G. Abramowitz, and H. Gupta. Are we unnecessarily constraining the agility of complex process-based models? *Water Resources Research*, 51(1):716–728, 2015. doi: https://doi.org/10.1002/2014WR015820. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014WR015820.

C. Meng, Y. Song, J. Song, and S. Ermon. Gaussianization flows. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 4336—-4345. PMLR, mar 2020a. URL http://proceedings.mlr.press/v108/meng20b.html.

C. Meng, Y. Song, J. Song, and S. Ermon. Gaussianization flows. *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, 108:4336–4345, 2020b. URL http://proceedings.mlr.press/v108/meng20b.html.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.

T. P. Minka. Expectation propagation for approximate bayesian inference. In J. S. Breese and D. Koller, editors, *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001*, pages 362–369. Morgan Kaufmann, 2001. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=120&proceeding_id=17.

P. J. Moreno, P. Ho, and N. Vasconcelos. A kullback-leibler divergence based kernel for SVM classification in multimedia applications. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 1385–1392. MIT Press, 2003. URL http://papers.nips.cc/paper/2351-a-kullback-leibler-divergence-based-kernel-for-svm-classification-in-multimedia-appl

M. Morris. Factorial sampling plans for preliminary computational experiments. *Quality Engineering*, 37:307–310, 1991.

K. Muandet, K. Fukumizu, B. K. Sriperumbudur, and B. Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Found. Trends Mach. Learn.*, 10(1-2):1–141, 2017. doi: 10.1561/2200000060. URL https://doi.org/10.1561/2200000060.

K. P. Murphy. *Machine learning - a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, 2012. ISBN 0262018020.

I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent gaussian models. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 1732–1740. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/4114-slice-sampling-covariance-hyperparameters-of-latent-gaussian-models.

E. T. Nalisnick, A. Matsukawa, Y. W. Teh, D. Görür, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=H1xwNhCcYm.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.

G. S. Nearing, Y. Tian, H. V. Gupta, M. P. Clark, K. W. Harrison, and S. V. Weijs. A philosophical basis for hydrological uncertainty. *Hydrological Sciences Journal*, 61(9): 1666–1678, 2016. doi: 10.1080/02626667.2016.1183009. URL https://doi.org/10.1080/02626667.2016.1183009.

A. M. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 427–436. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298640. URL https://doi.org/10.1109/CVPR.2015.7298640.

D. Nielsen, P. Jaini, E. Hoogeboom, O. Winther, and M. Welling. Survae flows: Surjections to bridge the gap between vaes and flows. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/9578a63fbe545bd82cc5bbe749636af1-Abstract.html.

R. Novak, L. Xiao, Y. Bahri, J. Lee, G. Yang, J. Hron, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1g30j0qF7.

R. Novak, L. Xiao, J. Hron, J. Lee, A. A. Alemi, J. Sohl-Dickstein, and S. S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020. URL https://github.com/google/neural-tangents.

J. Oakley and A. O'Hagan. Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, 89(4):769–784, 12 2002. ISSN 0006-3444. doi: 10.1093/biomet/89.4.769. URL https://doi.org/10.1093/biomet/89.4.769.

J. E. Oakley and A. OHagan. Probabilistic sensitivity analysis of complex models: a bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004. doi: https://doi.org/10.1111/j.1467-9868.2004.05304.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2004.05304.x.

Objax Developers. Objax, 2020. URL https://github.com/google/objax.

J. B. Oliva, A. Dubey, M. Zaheer, B. Póczos, R. Salakhutdinov, E. P. Xing, and J. Schneider. Transformation autoregressive networks. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning*

Research, pages 3895–3904. PMLR, 2018. URL http://proceedings.mlr.press/v80/oliva18a.html.

R. O'Neill and B. Rust. Aggregation error in ecological models. *Ecological Modelling*, 7 (2):91 – 105, 1979. ISSN 0304-3800. doi: https://doi.org/10.1016/0304-3800(79)90001-2. URL http://www.sciencedirect.com/science/article/pii/0304380079900012.

G. Papamakarios, I. Murray, and T. Pavlakou. Masked autoregressive flow for density estimation. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2338–2347, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6c1da886822c67822bcf3679d04369fa-Abstract.html.

G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference, 2019a. URL http://arxiv.org/abs/1912.02762.

G. Papamakarios, E. T. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *CoRR*, abs/1912.02762, 2019b. URL http://arxiv.org/abs/1912.02762.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. URL http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.

S. Paul, C. Boutsidis, M. Magdon-Ismail, and P. Drineas. Random projections for linear support vector machines. *ACM Trans. Knowl. Discov. Data*, 8(4):22:1–22:25, 2014. doi: 10.1145/2641760. URL https://doi.org/10.1145/2641760.

S. N. Payrovnaziri, Z. Chen, P. Rengifo-Moreno, T. Miller, J. Bian, J. H. Chen, X. Liu, and Z. He. Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review. *J. Am. Medical Informatics Assoc.*, 27(7): 1173–1185, 2020. doi: 10.1093/jamia/ocaa053. URL https://doi.org/10.1093/jamia/ocaa053.

C. A. Petri. Fundamentals of a theory of asynchronous information flow. In *Information Processing, Proceedings of the 2nd IFIP Congress 1962, Munich, Germany, August 27 - September 1, 1962*, pages 386–390. North-Holland, 1962.

D. Phan, N. Pradhan, and M. Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *CoRR*, abs/1912.11554, 2019. URL http://arxiv.org/abs/1912.11554.

F. Pianosi, K. Beven, J. Freer, J. W. Hall, J. Rougier, D. B. Stephenson, and T. Wagener. Sensitivity analysis of environmental models: A systematic review with practical workflow, may 2016. ISSN 13648152.

G. Pleiss, M. Jankowiak, D. Eriksson, A. Damle, and J. R. Gardner. Fast matrix square roots with applications to gaussian processes and bayesian optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fcf55a303b71b84d326fb1d06e332a26-Abstract.html.

R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2): 495–503, 11 2006. ISSN 0956-540X. doi: 10.1111/j.1365-246X.2006.02978.x. URL https://doi.org/10.1111/j.1365-246X.2006.02978.x.

C. Prieur, L. Viry, E. Blayo, and J.-M. Brankart. A global sensitivity analysis approach for marine biogeochemical modeling. *Ocean Modelling*, 139:101402, 2019. ISSN 1463-5003. doi: https://doi.org/10.1016/j.ocemod.2019.101402. URL https://www.sciencedirect.com/science/article/pii/S1463500318303688.

P. R. Raamana. Kernel methods library for pattern analysis and machine learning in python. *CoRR*, abs/2005.13483, 2020. URL https://arxiv.org/abs/2005.13483.

C. Rackauckas, Y. Ma, V. Dixit, X. Guo, M. Innes, J. Revels, J. Nyberg, and V. Ivaturi. A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. *CoRR*, abs/1812.01892, 2018. URL http://arxiv.org/abs/1812.01892.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1177–1184. Curran Associates, Inc., 2007. URL http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.

M. Raissi. Parametric gaussian process regression for big data. *CoRR*, abs/1704.03144, 2017. URL http://arxiv.org/abs/1704.03144.

A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9(83):2491–2521, 2008. URL http://jmlr.org/papers/v9/rakotomamonjy08a.html.

O. Rakovec, M. C. Hill, M. P. Clark, A. H. Weerts, A. J. Teuling, and R. Uijlenhoet. Distributed evaluation of local sensitivity analysis (DELSA), with application to hydrologic models. *Water Resources Research*, 50(1):409–426, 2014. ISSN 00431397. doi: 10.1002/2013WR014063.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X.

P. M. Rasmussen, K. H. Madsen, T. Lund, and L. Hansen. Visualization of nonlinear kernel models in neuroimaging by sensitivity maps. *NeuroImage*, 55:1120–1131, 2011.

S. Razavi and H. Gupta. What do we mean by sensitivity analysis? the need for comprehensive characterization of "global" sensitivity in earth and environmental systems models. *Water Resources Research*, 51:3070–3092, 2015a.

S. Razavi and H. V. Gupta. What do we mean by sensitivity analysis? the need for comprehensive characterization of "global" sensitivity in Earth and Environmental systems models. *Water Resources Research*, 51(5):3070–3092, may 2015b. ISSN 19447973. doi: 10.1002/2014WR016527.

S. Razavi, R. Sheikholeslami, H. V. Gupta, and A. Haghnegahdar. VARS-TOOL: A toolbox for comprehensive, efficient, and robust sensitivity and uncertainty analysis. *Environmental Modelling and Software*, 112(May 2018):95–107, 2019. ISSN 13648152. doi: 10.1016/j.envsoft.2018.10.005. URL https://doi.org/10.1016/j.envsoft.2018.10.005.

S. Razavi, A. Jakeman, A. Saltelli, C. Prieur, B. Iooss, E. Borgonovo, E. Plischke, S. Lo Piano, T. Iwanaga, W. Becker, S. Tarantola, J. H. Guillaume, J. Jakeman, H. Gupta, N. Melillo, G. Rabitti, V. Chabridon, Q. Duan, X. Sun, S. Smith, R. Sheikholeslami, N. Hosseini, M. Asadzadeh, A. Puy, S. Kucherenko, and H. R. Maier. The Future of Sensitivity Analysis: An Essential Discipline for Systems Modeling and Policy Support. *Environmental Modelling Software*, page 104954, dec 2020. ISSN 13648152. doi: 10.1016/j.envsoft.2020.104954. URL https://linkinghub.elsevier.com/retrieve/pii/S1364815220310112.

M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566:195–204, 2019.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1530–1538. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/rezende15.html.

D. J. Rezende, G. Papamakarios, S. Racanière, M. S. Albergo, G. Kanwar, P. E. Shanahan, and K. Cranmer. Normalizing flows on tori and spheres. In *Proceedings of the 37th*

International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, pages 8083–8092. PMLR, 2020. URL http://proceedings.mlr.press/v119/rezende20a.html.

J. P. Rivera, J. Verrelst, J. Gómez-Dans, J. Muñoz-Marí, J. F. Moreno, and G. Camps-Valls. An emulator toolbox to approximate radiative transfer models with statistical learning. *Remote. Sens.*, 7(7):9347–9370, 2015. doi: 10.3390/rs70709347. URL https://doi.org/10.3390/rs70709347.

P. Robert and Y. Escoufier. A unifying tool for linear multivariate statistical methods: The rv- coefficient. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 25(3): 257–265, 1976. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2347233.

J. L. Rojo-Alvarez, M. Martinez-Ramon, J. Munoz-Mari, and G. Camps-Valls. *Digital Signal Processing with Kernel Methods*. Wiley-IEEE Press, 1st edition, 2018. ISBN 1118611799.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2018.

H. Salimbeni and M. P. Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4588–4599, 2017. URL http://papers.nips.cc/paper/7045-doubly-stochastic-variational-inference-for-deep-gaussian-processes.

H. Salimbeni, S. Eleftheriadis, and J. Hensman. Natural gradients in practice: Non-conjugate variational inference in gaussian process models. In A. J. Storkey and F. Pérez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 689–697. PMLR, 2018. URL http://proceedings.mlr.press/v84/salimbeni18a.html.

A. Saltelli. A short comment on statistical versus mathematical modelling. *Nature Communications*, 10, 2019.

A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer*. Wiley, 2008. ISBN 9780470725177. URL https://books.google.es/books?id=wAssmt2vumgC.

J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Comput. Sci.*, 2:e55, 2016. doi: 10.7717/peerj-cs.55. URL https://doi.org/10.7717/peerj-cs.55.

W. Samek, G. Montavon, A. Vedaldi, L. Hansen, and K. Müller. Explainable ai: Interpreting, explaining and visualizing deep learning. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 2019.

D. L. Sánchez, J. M. Corchado, and A. G. Arrieta. Data-independent random projections from the feature-map of the homogeneous polynomial kernel of degree two. *Inf. Sci.*, 436-437:214–226, 2018. doi: 10.1016/j.ins.2018.01.022. URL https://doi.org/10.1016/j.ins.2018.01.022.

A. D. Saul, J. Hensman, A. Vehtari, and N. D. Lawrence. Chained gaussian processes. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 1431–1440. JMLR.org, 2016. URL http://proceedings.mlr.press/v51/saul16.html.

B. Schölkopf and A. J. Smola. A short introduction to learning with kernels. In S. Mendelson and A. J. Smola, editors, *Advanced Lectures on Machine Learning, Machine Learning Summer School 2002, Canberra, Australia, February 11-22, 2002, Revised Lectures*, volume 2600 of *Lecture Notes in Computer Science*, pages 41–64. Springer, 2002a. doi: 10.1007/3-540-36434-X_2. URL https://doi.org/10.1007/3-540-36434-X_2.

B. Schölkopf and A. J. Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002b. ISBN 9780262194754. URL https://www.worldcat.org/oclc/48970254.

B. Schölkopf, A. J. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998. doi: 10.1162/089976698300017467. URL https://doi.org/10.1162/089976698300017467.

A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Zídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis. Improved protein structure prediction using potentials from deep learning. *Nat.*, 577(7792):706–710, 2020. doi: 10.1038/s41586-019-1923-7. URL https://doi.org/10.1038/s41586-019-1923-7.

A. Shah, A. G. Wilson, and Z. Ghahramani. Student-t processes as alternatives to gaussian processes. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Workshop and Conference Proceedings*, pages 877–885. JMLR.org, 2014. URL http://proceedings.mlr.press/v33/shah14.html.

T. Shaikhina and N. A. Khovanova. Handling limited datasets with neural networks in medical applications: A small-data approach. *Artif. Intell. Medicine*, 75:51–63, 2017. doi: 10.1016/j.artmed.2016.12.003. URL https://doi.org/10.1016/j.artmed.2016.12.003.

C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(4):623–656, 1948. doi: 10.1002/j.1538-7305.1948.tb00917.x. URL https://doi.org/10.1002/j.1538-7305.1948.tb00917.x.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. ISBN 9780511809682. doi: 10.1017/CBO9780511809682. URL https://kernelmethods.blogs.bristol.ac.uk/.

R. Sheikholeslami, S. Razavi, and A. Haghnegahdar. What should we do when a model crashes? recommendations for global sensitivity analysis of earth and environmental systems models. *Geoscientific Model Development*, 12(10):4275–4296, 2019. doi: 10.5194/gmd-12-4275-2019. URL https://gmd.copernicus.org/articles/12/4275/2019/.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR, 2017. URL http://proceedings.mlr.press/v70/shrikumar17a.html.

R. Shwartz-Ziv and A. A. Alemi. Information in infinite ensembles of infinitely-wide neural networks. In C. Zhang, F. J. R. Ruiz, T. D. Bui, A. B. Dieng, and D. Liang, editors, *Symposium on Advances in Approximate Bayesian Inference, AABI 2019, Vancouver, BC, Canada, December 8, 2019*, volume 118 of *Proceedings of Machine Learning Research*, pages 1–17. PMLR, 2019. URL http://proceedings.mlr.press/v118/shwartz-ziv20a.html.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6034.

S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás. 3d deep learning on medical images: A review. *Sensors*, 20(18):5097, 2020. doi: 10.3390/s20185097. URL https://doi.org/10.3390/s20185097.

V. P. Singh. The use of entropy in hydrology and water resources. *Hydrological Processes*, 11(6):587–626, 1997. doi: https://doi.org/10.1002/(SICI)1099-1085(199705)11:6<587::AID-HYP479>3.0.CO;2-P. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-1085%28199705%2911%3A6%3C587%3A%3AAID-HYP479%3E3.0.CO%3B2-P.

D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL http://arxiv.org/abs/1706.03825.

E. Snelson and Z. Ghahramani. Local and global sparse gaussian process approximations. In M. Meila and X. Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS 2007, San Juan, Puerto Rico, March 21-24, 2007*, volume 2 of *JMLR Proceedings*, pages 524–531. JMLR.org, 2007. URL http://proceedings.mlr.press/v2/snelson07a.html.

E. Snelson, C. E. Rasmussen, and Z. Ghahramani. Warped gaussian processes. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 337–344. MIT Press, 2003. URL http://papers.nips.cc/paper/2481-warped-gaussian-processes.

H. Snoussi. Learning in presence of input noise using the stochastic EM algorithm. (2): 135–149, 2003. doi: 10.1063/1.1570540.

I. M. Sobolá. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 55:271–280, 2001.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *J. Mach. Learn. Res.*, 7:1531–1565, 2006. URL http://jmlr.org/papers/v7/sonnenburg06a.html.

D. Spiegelman, R. Logan, and D. Grove. Regression calibration with heteroscedastic error variance. *International Journal of Biostatistics*, 7(1), 2011. ISSN 15574679. doi: 10.2202/1557-4679.1259.

M. Studený and J. Vejnarová. The multiinformation function as a tool for measuring stochastic dependence. In M. I. Jordan, editor, *Learning in Graphical Models*, volume 89 of *NATO ASI Series*, pages 261–297. Springer Netherlands, 1998. doi: 10.1007/978-94-011-5014-9_10. URL https://doi.org/10.1007/978-94-011-5014-9_10.

B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Syst. Saf.*, 93(7):964–979, 2008. doi: 10.1016/j.ress.2007.04.002. URL https://doi.org/10.1016/j.ress.2007.04.002.

T. Sullivan. *Introduction to Uncertainty Quantification*. Texts in Applied Mathematics. Springer International Publishing, 2015. ISBN 9783319233956. URL https://books.google.es/books?id=Sik3CwAAQBAJ.

A. Y. Sun, D. Wang, and X. Xu. Monthly streamflow forecasting using gaussian process regression. *Journal of Hydrology*, 511:72–81, 2014.

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 2017. URL http://proceedings.mlr.press/v70/sundararajan17a.html.

D. H. Svendsen, L. Martino, and G. Camps-Valls. Active emulation of computer codes with gaussian processes – application to remote sensing. *Pattern Recognition*, 100:107103, 2020. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.107103. URL https://www.sciencedirect.com/science/article/pii/S0031320319304042.

A. Svensson, J. Dahlin, and T. B. Schön. Marginalizing gaussian process hyperparameters using sequential monte carlo. In *6th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2015, Cancun, Mexico, December 13-16, 2015*, pages 477–480. IEEE, 2015. doi: 10.1109/CAMSAP.2015.7383840. URL https://doi.org/10.1109/CAMSAP.2015.7383840.

G. J. Székely, M. L. Rizzo, and N. K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007. ISSN 00905364. URL http://www.jstor.org/stable/25464608.

E. G. Tabak and C. V. Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013. doi: https://doi.org/10.1002/cpa.21423. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21423.

S. Tang, W. J. Maddox, C. Dickens, T. Diethe, and A. C. Damianou. Similarity of neural networks with gradients. *CoRR*, abs/2003.11498, 2020. URL https://arxiv.org/abs/2003.11498.

S. Tennøe, G. Halnes, and G. T. Einevoll. Uncertainpy: A Python Toolbox for Uncertainty Quantification and Sensitivity Analysis in Computational Neuroscience. *Frontiers in Neuroinformatics*, 12, aug 2018. ISSN 16625196. doi: 10.3389/fninf.2018.00049.

N. M. Timme and C. Lapish. A tutorial for information theory in neuroscience. *eNeuro*, 5, 2018.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999. ISSN 13697412, 14679868. URL http://www.jstor.org/stable/2680726.

M. Titsias and N. Lawrence. Bayesian Gaussian Process Latent Variable Model. *Artificial Intelligence*, 9:844–851, 2010. ISSN 0899-7667. doi: 10.1162/089976699300016331. URL http://eprints.pascal-network.org/archive/00006343/.

M. K. Titsias, N. D. Lawrence, and M. Rattray. Efficient sampling for gaussian process inference using control variables. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1681–1688. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3414-efficient-sampling-for-gaussian-process-inference-using-control-variables.

S. Tokui, R. Okuta, T. Akiba, Y. Niitani, T. Ogawa, S. Saito, S. Suzuki, K. Uenishi, B. Vogel, and H. Y. Vincent. Chainer: A deep learning framework for accelerating the research cycle. In A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2002–2011. ACM, 2019. doi: 10.1145/3292500.3330756. URL https://doi.org/10.1145/3292500.3330756.

J. M. Tomczak and M. Welling. Improving variational auto-encoders using householder flow. *CoRR*, abs/1611.09630, 2016. URL http://arxiv.org/abs/1611.09630.

D. Tran, R. Ranganath, and D. M. Blei. Variational gaussian process. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06499.

M. G. Turner and R. H. Gardner. *Introduction to Models*, pages 63–95. Springer New York, New York, NY, 2015. ISBN 978-1-4939-2794-4. doi: 10.1007/978-1-4939-2794-4_3. URL https://doi.org/10.1007/978-1-4939-2794-4_3.

R. van den Berg, L. Hasenclever, J. M. Tomczak, and M. Welling. Sylvester normalizing flows for variational inference. In A. Globerson and R. Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 393–402. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/156.pdf.

G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller. Scikit-learn: Machine learning without learning the machinery. *GetMobile Mob. Comput. Commun.*, 19(1):29–33, 2015. doi: 10.1145/2786984.2786995. URL https://doi.org/10.1145/2786984.2786995.

C. Villacampa-Calvo, B. Zaldívar, E. C. Garrido-Merchán, and D. Hernández-Lobato. Multi-class Gaussian process classification with noisy inputs, 2020. ISSN 23318422. URL http://arxiv.org/abs/2001.10523.

D. Vuuren, J. Edmonds, M. Kainuma, K. Riahi, A. Thomson, K. Hibbard, G. Hurtt, T. Kram, V. Krey, J. Lamarque, T. Masui, M. Meinshausen, N. Nakicenovic, S. Smith, and S. Rose. The representative concentration pathways: an overview. *Climatic Change*, 109:5–31, 2011.

C. Wang and D. M. Blei. Variational inference in nonconjugate models. *J. Mach. Learn. Res.*, 14(1):1005–1031, 2013. URL http://dl.acm.org/citation.cfm?id=2502613.

K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Wang. Generative adversarial networks: introduction and outlook. *IEEE CAA J. Autom. Sinica*, 4(4):588–598, 2017. doi: 10.1109/JAS.2017.7510583. URL https://doi.org/10.1109/JAS.2017.7510583.

K. A. Wang, G. Pleiss, J. R. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson. Exact gaussian processes on a million data points. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14622–14632, 2019a. URL http://papers.nips.cc/paper/9606-exact-gaussian-processes-on-a-million-data-points.

P. Wang, Z. Lu, and S. Xiao. Variance-based sensitivity analysis with the uncertainties of the input variables and their distribution parameters. *Commun. Stat. Simul. Comput.*, 47 (4):1103–1125, 2018. doi: 10.1080/03610918.2017.1307394. URL https://doi.org/10.1080/03610918.2017.1307394.

W. Wang, Y. Huang, Y. Wang, and L. Wang. Generalized autoencoder: A neural network framework for dimensionality reduction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014*, pages 496–503. IEEE Computer Society, 2014. doi: 10.1109/CVPRW.2014.79. URL https://doi.org/10.1109/CVPRW.2014.79.

Y. Wang, H. F. Tung, A. J. Smola, and A. Anandkumar. Fast and guaranteed tensor decomposition via sketching. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 991–999, 2015. URL http://papers.nips.cc/paper/5944-fast-and-guaranteed-tensor-decomposition-via-sketching.

Y. Wang, G. Wei, and D. Brooks. Benchmarking tpu, gpu, and CPU platforms for deep learning. *CoRR*, abs/1907.10701, 2019b. URL http://arxiv.org/abs/1907.10701.

M. S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM J. Res. Dev.*, 4(1):66–82, 1960. doi: 10.1147/rd.41.0066. URL https://doi.org/10.1147/rd.41.0066.

O. Watt-Meyer, N. D. Brenowitz, S. K. Clark, B. Henn, A. Kwa, J. J. McGibbon, W. A. Perkins, and C. S. Bretherton. Correcting weather and climate models by machine learning nudged historical simulations. *Earth and Space Science Open Archive*, page 13, 2021. doi: 10.1002/essoar.10505959.1. URL https://doi.org/10.1002/essoar.10505959.1.

S. Weijs, G. Schoups, and N. Giesen. Why hydrological predictions should be evaluated using information theory. *Hydrology and Earth System Sciences*, 14:2545–2558, 2010.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 681–688. Omnipress, 2011. URL https://icml.cc/2011/papers/398_icmlpaper.pdf.

F. Wenzel, K. Roth, B. S. Veeling, J. Swiatkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin. How good is the bayes posterior in deep neural networks really? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10248–10259. PMLR, 2020. URL http://proceedings.mlr.press/v119/wenzel20a.html.

J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar. Integrating physics-based modeling with machine learning: A survey. *ArXiv*, abs/2003.04919, 2020.

C. K. I. Williams and M. W. Seeger. Using the nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 682–688. MIT Press, 2000. URL http://papers.nips.cc/paper/1866-using-the-nystrom-method-to-speed-up-kernel-machines.

A. G. Wilson and R. P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1067–1075. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/wilson13.html.

A. G. Wilson and P. Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *CoRR*, abs/2002.08791, 2020. URL https://arxiv.org/abs/2002.08791.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Stochastic variational deep kernel learning. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2586–2594, 2016a. URL http://papers.nips.cc/paper/6426-stochastic-variational-deep-kernel-learning.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 370–378. JMLR.org, 2016b. URL http://proceedings.mlr.press/v51/wilson16.html.

J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Pathwise conditioning of gaussian processes. *CoRR*, abs/2011.04026, 2020. URL https://arxiv.org/abs/2011.04026.

H. Wu, J. Köhler, and F. Noé. Stochastic normalizing flows. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*

2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/41d80bfc327ef980528426fc810a6d7a-Abstract.html.

L. Xiao, J. Pennington, and S. S. Schoenholz. Disentangling trainability and generalization in deep learning. *CoRR*, abs/1912.13053, 2019. URL http://arxiv.org/abs/1912.13053.

C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *CoRR*, abs/1304.5634, 2013. URL http://arxiv.org/abs/1304.5634.

G. Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *CoRR*, abs/1902.04760, 2019. URL http://arxiv.org/abs/1902.04760.

G. Yang, T. Zhang, P. Kirichenko, J. Bai, A. G. Wilson, and C. D. Sa. SWALP : Stochastic weight averaging in low precision training. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7015–7024. PMLR, 2019. URL http://proceedings.mlr.press/v97/yang19d.html.

T. Yang, Y. Li, M. Mahdavi, R. Jin, and Z. Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 485–493, 2012. URL http://papers.nips.cc/paper/4588-nystrom-method-vs-random-fourier-features-a-theoretical-and-empirical-comparison.

Y. Yang and P. Perdikaris. Adversarial uncertainty quantification in physics-informed neural networks. *J. Comput. Phys.*, 394:136–152, 2019. doi: 10.1016/j.jcp.2019.05.027. URL https://doi.org/10.1016/j.jcp.2019.05.027.

T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Comput. Intell. Mag.*, 13(3):55–75, 2018. doi: 10.1109/MCI.2018.2840738. URL https://doi.org/10.1109/MCI.2018.2840738.

F. X. Yu, A. T. Suresh, K. M. Choromanski, D. N. Holtmann-Rice, and S. Kumar. Orthogonal random features. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1975–1983, 2016. URL http://papers.nips.cc/paper/6246-orthogonal-random-features.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Q. Zhang and Y. Ni. Improved most likely heteroscedastic gaussian process regression via bayesian residual moment estimator. *IEEE Trans. Signal Process.*, 68:3450–3460, 2020. doi: 10.1109/TSP.2020.2997940. URL https://doi.org/10.1109/TSP.2020.2997940.

X. Zhang and S. Liao. Incremental randomized sketching for online kernel learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7394–7403. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhang19h.html.

D. Annex: Scientific Publications

Paper I

J. E. Johnson, V. Laparra, A. Pérez-Suay, M. D. Mahecha, G. Camps-Valls, "Kernel methods and their derivatives: Concept and perspectives for the earth system sciences", in PLOS ONE15(10): e0235885, October2020, doi:10.1371/journal.pone.0235885.
This journal has an impact factor of 2.74 and within the first quartile in several categories: Q1: Electrical and Electronic Engineering, Q1: Biology, Q1: Imaging Science.

Paper II

J. E. Johnson, V. Laparra and G. Camps-Valls, "Accounting for Input Noise in Gaussian Process Parameter Retrieval", in IEEE Geoscience and Remote Sensing Letters, vol.17, no.3, pp.391-395, March2020, doi:10.1109/LGRS.2019.2921476.
This journal has an impact factor of 3.83 and within the first quartile in several categories: Q1: Remote Sensing, Q1: Electrical and Electronic Engineering

Paper III

J. E. Johnson, V. Laparra, Maria Piles, and G. Camps-Valls, "Gaussianizing the Earth: Multidimensional Information Measures for Earth Data Analysis", in IEEE Geoscience and Remote Sensing Magazine,2021(Accepted).
This journal has an impact factor of 13.0 and within the first quartile in several categories: Q1: Geochemistry and Geophysics, Q1: Remote Sensing.

D.1. Paper I

PLOS ONE

RESEARCH ARTICLE

Kernel methods and their derivatives: Concept and perspectives for the earth system sciences

J. Emmanuel Johnson*, Valero Laparra, Adrián Pérez-Suay, Miguel D. Mahecha¤, Gustau Camps-Valls

Image Processing Laboratory, Universitat de València, València, Spain

¤ Current address: Max Planck Institute for Biogeochemistry, Jena, Germany
* juan.johnson@uv.es

OPEN ACCESS

Citation: Johnson JE, Laparra V, Pérez-Suay A, Mahecha MD, Camps-Valls G (2020) Kernel methods and their derivatives: Concept and perspectives for the earth system sciences. PLoS ONE 15(10): e0235885. https://doi.org/10.1371/journal.pone.0235885

Editor: Qichun Zhang, University of Bradford, UNITED KINGDOM

Received: July 10, 2019

Accepted: June 8, 2020

Published: October 29, 2020

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0235885

Data Availability Statement: All toy example code is reproducible and is available at: https://github.com/IPL-UV/sakame All applied data is open

Abstract

Kernel methods are powerful machine learning techniques which use generic non-linear functions to solve complex tasks. They have a solid mathematical foundation and exhibit excellent performance in practice. However, kernel machines are still considered black-box models as the kernel feature mapping cannot be accessed directly thus making the kernels difficult to interpret. The aim of this work is to show that it is indeed possible to interpret the functions learned by various kernel methods as they can be intuitive despite their complexity. Specifically, we show that derivatives of these functions have a simple mathematical formulation, are easy to compute, and can be applied to various problems. The model function derivatives in kernel machines is proportional to the kernel function derivative and we provide the explicit analytic form of the first and second derivatives of the most common kernel functions with regard to the inputs as well as generic formulas to compute higher order derivatives. We use them to analyze the most used supervised and unsupervised kernel learning methods: Gaussian Processes for regression, Support Vector Machines for classification, Kernel Entropy Component Analysis for density estimation, and the Hilbert-Schmidt Independence Criterion for estimating the dependency between random variables. For all cases we expressed the derivative of the learned function as a linear combination of the kernel function derivative. Moreover we provide intuitive explanations through illustrative toy examples and show how these same kernel methods can be applied to applications in the context of spatio-temporal Earth system data cubes. This work reflects on the observation that function derivatives may play a crucial role in kernel methods analysis and understanding.

1 Introduction

Kernel methods (KMs) constitute a standard set of tools in machine learning and pattern analysis [1, 2]. They are based on a mathematical framework to cope with nonlinear problems while still relying on well-established concepts of linear algebra. KMs are one of the preferred

source and available at: https://www.earthsystemdatalab.net/.

Funding: GCV 647423 European Research Council https://erc.europa.eu/ The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

tools in applied sciences, from signal and image processing [3], to computer vision [4] and geosciences [5]. Since its introduction in the 1990s through the popular support vector machines (SVMs), kernel methods have evolved into a large family of techniques that cope with many problems in addition to classification. Kernel machines have also excelled in regression, interpolation and function approximation problems [3], where Gaussian Processes (GPs) [6] and support vector regression [7] have provided good results in many applications. Furthermore, many kernel methods have been engineered to deal with other relevant learning problems; for example, density estimation via kernel decompositions using entropy components [8]. For dimensionality reduction and feature extraction, there are a wide family of multivariate data analysis kernel methods such as kernel principal component analysis [9], kernel canonical analysis [10] or kernel partial least squares [11]. Kernels have also been exploited to estimate dependence (nonlinear associations) between random variables such as kernel mutual information [12], or the Hilbert-Schmidt Independence Criterion [13]. Finally in the literature, we find kernel machines for data sorting [14], manifold learning and alignment [15], system identification [16], signal deconvolution and blind source separation [3].

However, *understanding* a model is more difficult than just *applying* a model, and kernel methods are still considered black-box models. Little can be said about the characteristics of the feature mapping which is only implicit in the formulation. Several approaches have been presented in the literature to *explore* the kernel feature mapping and to understand what the kernel machine is actually learning. One way to analyze kernel machines is by visualizing the empirical feature maps but this is very challenging and only feasible in low-dimensional problems [1, 17]. Another approach is to study the relative relevance of the input features (covariates) on the output. This is commonly referred to as feature ranking and it typically reduces to evaluating how the function varies when an input is removed or perturbed. Automatic relevance determination (ARD) kernels [6] or multiple kernel learning [18] allow one to study the relevance of the feature components indirectly. While this approach has been extensively used to improve the accuracy and understanding of supervised kernel classifiers and regression methods, they only provide feature ranking and nothing is said about the geometrical properties of the feature map. In order to resolve this, two main approaches are available in the kernel methods literature. For some particular kernels one can derive the metric induced by the kernel to give insight into the surfaces and structures [19]. Alternatively, one can study the feature map (in physically meaningful units) by learning the inverse feature mapping; a group of techniques known as kernel pre-imaging [20, 21]. However, the current methods are computationally expensive, involve critical parameters, and very often provide unstable results.

Function derivatives is a classical way to describe and visualize some characteristics of models. Derivatives of kernel functions have been introduced before, yet mostly used in supervised learning as a form of regularization that controls fast variations of the decision function [22]. However, derivatives of the model's function with regards to the input features for feature understanding and visualization has received less attention. A recent strategy is to derive sensitivity maps from a kernel feature map [23]. The sensitivity map is related to the squared derivative of the function with respect to the input features. The idea was originally derived for SVMs in neuroimaging applications [24], and later extended to GPs in geoscience problems [25–28]. In both cases, the goal was to retrieve a feature ranking from a learned supervised model.

In this paper, we analyze the kernel function derivatives for supervised and unsupervised kernel methods with several kernel functions in different machine learning paradigms. We show the usefulness of the derivatives to study and visualize kernel models in regression, classification, density estimation, and dependence estimation with kernels. Since differentiation is a linear operator, most kernel methods have a derivative that is proportional to the derivative of

the kernel function. We provide the analytic form of the first and second derivatives of the most common kernel functions with regards to the inputs, along with iterative formulas to compute the m-th order derivative of differentiable kernels, and for the radial basis function kernel in particular; where m is the number of successive derivatives. In classification problems, the derivatives can be related to the margin, and allow us to gain some insight on sampling [29]. In regression problems, a models' function derivatives may give insight about the signal and noise characteristics that allow one to design regularization functionals. In density estimation, the second derivative (the Hessian) allows us to follow the density ridge for manifold learning [30], whereas in dependence estimation squared derivatives (the sensitivity maps) allows one to study the most relevant points and features governing the association measure [31]. All in all, kernel derivatives allow us to identify both examples and features that affect the predictive function the most, and allow us to interpret the kernel model behavior in different learning applications. We show that the solutions can be expressed in closed-form for the most common kernel functions and kernel methods, they are easy to compute, and we give examples of how they can be used in practice.

The remainder of the paper is organized as follows. Section 2 briefly reviews the fundamentals of kernel functions and feature maps, and concentrates on the kernel derivatives for feature map analysis where we provide the first and second order derivatives for most of the common kernel functions. We also review the main ideas to summarize the information contained in the derivatives. Section 3 and Section 4 study popular discriminative kernel methods, such as Gaussian Processes for regression and support vector machines for classification. Section 5 analyzes the interesting case of density estimation with kernels, in particular through the use of kernel entropy component analysis for density estimation. Section 6 pays attention to the case of dependence estimation between random variables using the Hilbert-Schmidt independence criterion in cases of dependence visualization maps and data unfolding. Section 7 illustrates the applicability of kernel derivatives in the previous kernel methods on spatio-temporal Earth system science data. We conclude in section 8 with some final remarks.

2 Kernel functions and the derivatives

2.1 Kernel functions and feature maps

In this section, we briefly highlight the most important properties of kernel methods, needed to understand their role of the kernel methods mentioned in the subsequent sections. Recall that kernel methods rely on the notion of similarity between points in a higher (possibly infinite) dimensional Hilbert space. Let us consider a set of empirical data $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, whose elements are defined in a d-dimensional input space, $\boldsymbol{x}_i = [x_i^1, \ldots, x_i^d]^\top \in \mathbb{R}^d$, $1 \leq i \leq n$. In supervised settings, each input feature vector $\boldsymbol{x}$ is associated with a target value, which can be either discrete in the classification case, $y_i \in \mathbb{Z}^+$ or real in the regression case, $y_i \in \mathbb{R}$, $i = 1, \ldots, n$. Kernel methods assume the existence of a Hilbert space $\mathcal{H}$ with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ where samples in $\mathcal{X}$ are mapped into with a feature map $\phi : \mathcal{X} \to \mathcal{H}, \boldsymbol{x}_i \mapsto \phi(\boldsymbol{x}_i)$, $1 \leq i \leq n$. The mapping function can be defined explicitly (if some prior knowledge about the problem is available) or implicitly, which is often the case in kernel methods. The similarity between the elements in $\mathcal{H}$ can be estimated using its associated dot product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ via reproducing kernels in Hilbert spaces (RKHS), $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that pairs of points $(\boldsymbol{x}, \boldsymbol{x}') \mapsto k(\boldsymbol{x}, \boldsymbol{x}')$. So we can estimate similarities in $\mathcal{H}$ without the explicit definition of the *feature map* φ, and hence without having access to the points in $\mathcal{H}$. This *kernel function* k is required to satisfy Mercer's Theorem [32].

Definition 1 *Reproducing kernel Hilbert spaces (RKHS)* [33]. *A Hilbert space $\mathcal{H}$ is said to be a RKHS if: (1) The elements of $\mathcal{H}$ are complex or real valued functions $f(\cdot)$ defined on any set of elements $\boldsymbol{x}$; And (2) for every element $\boldsymbol{x}$, $f(\cdot)$ is bounded.*

The name of these spaces comes from the so-called *reproducing property*. In a RKHS $\mathcal{H}$, there exists a function $k(\cdot, \cdot)$ such that

$$f(\boldsymbol{x}) = \langle f, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}}, \quad f \in \mathcal{H}, \tag{1}$$

by virtue of the Riesz Representation Theorem [34]. In particular, for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{x}') \rangle_{\mathcal{H}} \tag{2}$$

A large class of algorithms have originated from regularization schemes in RKHS. The *representer theorem* gives us the general form of the solution to the common loss function formed by the loss term and a regularization term.

Theorem 1 (*Representer Theorem*) [34, 35] *Let $\Omega : [0, \infty) \to \mathbb{R}$ be a strictly monotonic increasing function; let $\mathbf{V} : (\mathcal{X} \times \mathbb{R}^2)^n \to \mathbb{R} \cup \{\infty\}$ be an arbitrary loss function; and let $\mathcal{H}$ be a RKHS with reproducing kernel k. Then*:

$$f^* = \min_{f \in \mathcal{H}} \{\mathbf{V}((\boldsymbol{x}_1, y_1, f(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_n, y_n, f(\boldsymbol{x}_n))) + \Omega(\| f \|^2_{\mathcal{H}})\} \tag{3}$$

admits a space of functions f defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i), \quad \alpha_i \in \mathbb{R}, \quad \alpha \in \mathbb{R}^n, \tag{4}$$

which is expressed as a linear combination of kernel functions. Also note that the previous theorem states that solutions imply having access to an empirical risk term $\mathbf{V}$ and a regularizer Ω. In the case of not having labels y_i, alternative representer theorems can be equally defined. A generalized representer theorem was introduced in [36], which generalizes Wahba's theorem to a larger class of regularizers and empirical losses. Also, in [37], a representer theorem for kernel principal components analysis (KPCA) was used: the theorem gives the solution as a linear combination of kernel functions centered at the input data points, and is called the representer theorem of learning theory [38], whereby the coefficients are determined by the eigendecomposition of the kernel matrix [9, 36]. Should the reader want more literature related to kernel methods, we highly recommend this paper [39] for a more theoretical introduction to Hilbert-Spaces in the context of kernel methods and [3] for a more applied and practical approaches.

2.2 Derivatives of linear expansions of kernel functions

Computing the derivatives of function f can give important insights about the learned model. Interestingly, in the majority of kernel methods, the function f is linear in the parameters $\boldsymbol{\alpha}$, cf. Eq (4) derived from the representer theorem [35] [Th. 1]. For the sake of simplicity, we will denote the partial derivative of f w.r.t. the feature x^j as $\partial_j f(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial x^j}$, where j denotes the dimension. This allows us to write the partial derivative of f as:

$$\partial_j f(\boldsymbol{x}) := \frac{\partial f(\boldsymbol{x})}{\partial x^j} = \frac{\partial \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i)}{\partial x^j} = \sum_{i=1}^{n} \alpha_i \frac{\partial k(\boldsymbol{x}, \boldsymbol{x}_i)}{\partial x^j} = (\partial_j \boldsymbol{k}(\boldsymbol{x}))^\top \boldsymbol{\alpha}, \tag{5}$$

where $\partial_j \boldsymbol{k}(\boldsymbol{x}) := \left[\frac{\partial k(\boldsymbol{x}, \boldsymbol{x}_1)}{\partial x^j}, \ldots, \frac{\partial k(\boldsymbol{x}, \boldsymbol{x}_n)}{\partial x^j}\right]^\top \in \mathbb{R}^n$ and $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^\top \in \mathbb{R}^n$. It is possible to take

the second order derivative with respect to feature x^j twice which remains linear as well with $\boldsymbol{\alpha}$:

$$\partial_j^2 f(\boldsymbol{x}) := \frac{\partial^2 f(\boldsymbol{x})}{\partial x^j \partial x^j} = \frac{\partial^2 \sum_i \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i)}{\partial x^{j2}} = \sum_i \alpha_i \frac{\partial^2 k(\boldsymbol{x}, \boldsymbol{x}_i)}{\partial x^{j2}} = (\partial_j^2 \boldsymbol{k}(\boldsymbol{x}))^\top \boldsymbol{\alpha}, \tag{6}$$

where $\partial_j^2 \boldsymbol{k}(\boldsymbol{x}) := \left[\frac{\partial^2 k(\boldsymbol{x}, \boldsymbol{x}_1)}{\partial x^{j2}}, \ldots, \frac{\partial^2 k(\boldsymbol{x}, \boldsymbol{x}_n)}{\partial x^{j2}}\right]^\top \in \mathbb{R}^n$. Inductively, the m-th partial derivative w.r.t the j-th feature is also linear with $\boldsymbol{\alpha}$ and it follows the following equation:

$$\partial_j^m f(\boldsymbol{x}) = (\partial_j^m \boldsymbol{k}(\boldsymbol{x}))^\top \boldsymbol{\alpha}. \tag{7}$$

The gradient of f gives information about the slope (increase rate) of the function and reduces to

$$\nabla f = \left[\frac{\partial f(\boldsymbol{x})}{\partial x^1}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial x^d}\right]^\top = (\nabla \mathbf{K})^\top \boldsymbol{\alpha} \in \mathbb{R}^d, \tag{8}$$

where ∇ denotes the vector differential operator, and $\nabla \mathbf{K} = [\partial_1 \boldsymbol{k}(\boldsymbol{x}) | \cdots | \partial_d \boldsymbol{k}(\boldsymbol{x})]$. The Laplacian accounts for the curvature, roughness, or concavity of the function itself, and can be easily computed as the sum of all the unmixed second partial derivatives, which for kernels reduces to

$$\nabla^2 f := \sum_{j=1}^{d} \frac{\partial^2 f(\boldsymbol{x})}{\partial x^{j2}} = \mathbb{1}_d^\top (\nabla^2 \mathbf{K})^\top \boldsymbol{\alpha} \in \mathbb{R}, \tag{9}$$

where $\nabla^2 \mathbf{K} = [\partial_1^2 \boldsymbol{k}(\boldsymbol{x}) | \cdots | \partial_d^2 \boldsymbol{k}(\boldsymbol{x})]$ and $\mathbb{1}_d$ is a column vector of ones of size d. Another useful descriptor is the Hessian matrix of f, which characterizes its local curvature. The Hessian is a $d \times d$ matrix of second-order partial derivatives with respect to the features x^j, x^k:

$$[\boldsymbol{H}]_{jk} = \frac{\partial^2 f(\boldsymbol{x})}{\partial x^j \partial x^k} = (\partial_j \partial_k \boldsymbol{k}(\boldsymbol{x}))^\top \boldsymbol{\alpha} \in \mathbb{R}. \tag{10}$$

The equations listed above have shown that the derivative of a kernel function is linear with $\boldsymbol{\alpha}$. Once the $\boldsymbol{\alpha}$ is computed, the problem reduces to (1) computing the derivatives for a particular kernel function, and (2) to summarize the information contained within the derivatives.

- **Derivatives of common kernel functions**. Kernel methods typically use a set of positive definite kernel functions, such as the linear, polynomial (Poly), hyperbolic tangent (Tanh), Gaussian (RBF) kernel, and the automatic relevance determination (ARD) kernel. We give the partial derivative for all of these kernels in Table 1, and the (mixed) second derivatives in Table 2. For the most widely used kernels (RBF and ARD), one can recognize a linear relation between the kernel derivative and the kernel function itself. It can be shown that the m-

Table 1. Partial derivatives for some common kernel functions: Linear, Polynomial (Poly), Radial Basis Functions (RBF), Hyperbolic tangent (Tanh), and Automatic Relevance Determination (ARD).

Kernel	Kernel function, $k(\boldsymbol{x}, \boldsymbol{y})$	Partial derivative, $\frac{\partial k(\boldsymbol{x}, \boldsymbol{y})}{\partial x^j}$
Linear	$\boldsymbol{x}^\top \boldsymbol{y}$	y^j
Poly	$(\gamma \boldsymbol{x}^\top \boldsymbol{y} + c_0)^p$	$\gamma p y^j (\gamma \boldsymbol{x}^\top \boldsymbol{y} + c_0)^{p-1}$
RBF	$\exp(-\gamma \lVert \boldsymbol{x} - \boldsymbol{y} \rVert^2)$	$-2\gamma(x^j - y^j) k(\boldsymbol{x}, \boldsymbol{y})$
Tanh	$\tanh(\gamma \boldsymbol{x}^\top \boldsymbol{y} + c_0)$	$\gamma y^j \operatorname{sech}^2(\gamma \boldsymbol{x}^\top \boldsymbol{y} + c_0)$
ARD	$v^2 \exp\left(-\frac{1}{2} \sum_{d=1}^{D} \left(\frac{x^d - y^d}{\lambda_d}\right)^2\right)$	$\left(\frac{x^j - y^j}{\lambda_j^2}\right) k(\boldsymbol{x}, \boldsymbol{y})$

https://doi.org/10.1371/journal.pone.0235885.t001

Table 2. Second derivatives for some common kernel functions.

Kernel	2nd partial derivative, $\frac{\partial^2 k(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}^j{}^2}$	Mixed partial derivative, $\frac{\partial^2 k(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}^j \partial \boldsymbol{x}^k}$
Linear	0	0
Poly	$(p-1)p\,(\gamma y^j)^2\,(\gamma\,\boldsymbol{x}^\top\boldsymbol{y}+c_0)^{p-2}$	$(p-1)p\gamma^2\,y^j\,y^k\,(\gamma\,\boldsymbol{x}^\top\boldsymbol{y}+c_0)^{p-2}$
RBF	$2\gamma\,[2\gamma\,(x^j-y^j)^2-1]\,k(\boldsymbol{x},\boldsymbol{y})$	$4\gamma^2\,(x^k-y^k)\,(x^j-y^j)\,k(\boldsymbol{x},\boldsymbol{y})$
Tanh	$-2(\gamma y^j)^2\,\mathrm{sech}^2(\gamma\,\boldsymbol{x}^\top\boldsymbol{y}+c_0)k(\boldsymbol{x},\boldsymbol{y})$	$-2\gamma^2\,y^j\,y^k\,\mathrm{sech}^2(\gamma\,\boldsymbol{x}^\top\boldsymbol{y}+c_0)k(\boldsymbol{x},\boldsymbol{y})$
ARD	$\left(\frac{1}{\lambda_j^2}+\left(\frac{x^j-y^j}{\lambda_j^2}\right)^2\right)k(\boldsymbol{x},\boldsymbol{y})$	$\left(\frac{x^j-y^j}{\lambda_j^2}\right)\left(\frac{x^k-y^k}{\lambda_k^2}\right)k(\boldsymbol{x},\boldsymbol{y})$

https://doi.org/10.1371/journal.pone.0235885.t002

th derivative of some kernel functions can be computed recursively using Faà di Bruno's identity [40].

- **Summarizing function derivatives**. Summarizing the information contained in the derivatives is not an easy task, especially in high dimensional problems. The most obvious strategy is to use the norm of the partial derivative, that is $\|\partial_j f\|$, which summarizes the relevance of variable x^j. A small norm implies a small change in the discriminative function f with respect to the j-th dimension, indicating the low importance of that feature. This approach was introduced as *sensitivity maps* (SMs) in [24] for the visualization of SVM maps in neuroimaging and later exploited in GPs for ranking spectral channels in geosciences applications [26]. The SM for the j-th feature, is the expected value of the squared derivative of the function with respect the input argument x^j:

$$s^j = \int_{\mathcal{X}^j} \left(\frac{\partial f(\boldsymbol{x})}{\partial x^j}\right)^2 p(x^j)\mathrm{d}x^j, \tag{11}$$

where $p(x)$ is the probability density function (pdf) over dimension j of the input space $\mathcal{X}$. In order to avoid the possibility of cancellation of the terms due to its signs, the derivatives are squared. Other transformations like the absolute value could be equally applied. The *empirical sensitivity map* approximation to Eq (11) is obtained by replacing the expected value with a summation over the available n samples

$$s^j \approx \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial f(\boldsymbol{x}_i)}{\partial x_i^j}\right)^2, \tag{12}$$

which can be grouped together to define the *sensitivity vector* as $\mathbf{s} = [s^1, \ldots, s^d]^\top$.
This can be thought of as studying the relevance of the sample points. Similarly,one can average over the features to obtain a *point sensitivity*:

$$q_i = \frac{1}{d}\sum_{j=1}^{d}\left(\frac{\partial f(\boldsymbol{x}_i)}{\partial x_i^j}\right)^2, \tag{13}$$

which can be grouped to define the *point sensitivity vector* as $\mathbf{q} = [q_1, \ldots, q_n]^\top$. The information contained in $\mathbf{q}$ is related to the robustness to changes of the decision in each point of the space.

Now we are equipped to use the derivatives and the corresponding sensitivity maps in arbitrary kernel machines that use standard kernel functions. In the following sections, we study

its use in kernel methods for both supervised (regression and classification) and unsupervised (density estimation and dependence estimation) learning.

3 Kernel regression

3.1 Gaussian Process Regression

Multiple proposals to use kernel methods in a regression framework have been done during the last few decades. Gaussian Processes (GPs) is perhaps the most successful kernel method for discriminative learning in general and regression in particular [6]. Standard GP regression approximates observations as the sum of some unknown latent function $f(\boldsymbol{x})$ of the inputs plus some additive Gaussian noise, $y_i = f(\boldsymbol{x}_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. A zero mean GP prior is placed on the latent function $f(\boldsymbol{x})$ and a Gaussian prior is used for each latent noise term ε_i, in other words $f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), \mathbf{K})$, where $m(\boldsymbol{x}) = 0$, and $\mathbf{K}$ is a covariance function, $[\mathbf{K}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, parameterized by a set of hyperparameters $\boldsymbol{\theta}$ (e.g. $\boldsymbol{\theta} = [\lambda_1, \ldots, \lambda_d]$ for the ARD kernel function).

If we consider a test location $\boldsymbol{x}_*$ with the corresponding output y_*, a $\mathcal{GP}$ prior induces a prior distribution between the observations $\boldsymbol{y}$ and y_*. Collecting all available data in $\mathcal{D} \equiv \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots n\}$, it is possible to analytically compute the posterior distribution over the unknown output y_* given the test input $\boldsymbol{x}_*$ and the available training set $\mathcal{D}$, $p(y_*|\boldsymbol{x}_*, \mathcal{D}) = \mathcal{N}(y_*|\mu_{\mathrm{GP}*}, \sigma^2_{\mathrm{GP}*})$, which is a Gaussian with the following mean and variance:

$$\mu_{\mathrm{GP}*} = \boldsymbol{k}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \boldsymbol{y} = \boldsymbol{k}_*^\top \boldsymbol{\alpha}, \tag{14}$$

$$\sigma^2_{\mathrm{GP}*} = \sigma_n^2 + k_{**} - \boldsymbol{k}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \boldsymbol{k}_*, \tag{15}$$

where $\boldsymbol{k}_* = \boldsymbol{k}(\boldsymbol{x}_*) = [k(\boldsymbol{x}_*, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}_*, \boldsymbol{x}_n)]^\top \in \mathbb{R}^n$ contains the kernel similarities of the test point $\boldsymbol{x}_*$ to all training points in $\mathcal{D}$, $\mathbf{K}$ is a $n \times n$ kernel (covariance) matrix whose entries contain the similarities between all training points, $\boldsymbol{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$, $k_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$ is a scalar with the self-similarity of $\boldsymbol{x}_*$, and $\boldsymbol{I}$ is the identity matrix. The solution of the predictive mean for the GP model in (14) is expressed in the same way as equation (4), where $\mu_{\mathrm{GP}*} = f(\boldsymbol{x}_*) = \boldsymbol{k}_*^\top \boldsymbol{\alpha}$. This expression is exactly the same as in other kernel regression methods like the Kernel Ridge Regression (KRR) [2] or the Relevance Vector Machine (RVM) [2]. The derivative of the mean function can be computed through Eq (5) and the derivatives in Table 1.

3.2 Derivatives and sensitivity maps

Let us start by visualizing derivatives in simple 1D examples. We used GP modeling with a standard RBF kernel function to fit five regression data sets. We show in Fig 1 the first and second derivatives of the fitted GP model, as well as the point-wise sensitivities. In all cases, first derivatives are related to positive or negative slopes, while the second derivatives are related to the curvature of the function. Since the derivative is a linear operator, a composition of functions is also the composition of derivatives as can be seen in the last two functions. This could be useful for analyzing more complex composite kernels. See Table 3 for a comparison with other kernel methods derivatives.

3.3 Derivatives and regularization

We show an example of applying the derivative of the kernel function as a regularization parameter for the noise. We modeled the function $f(x) = \sin(3\pi x)$ with an additive white

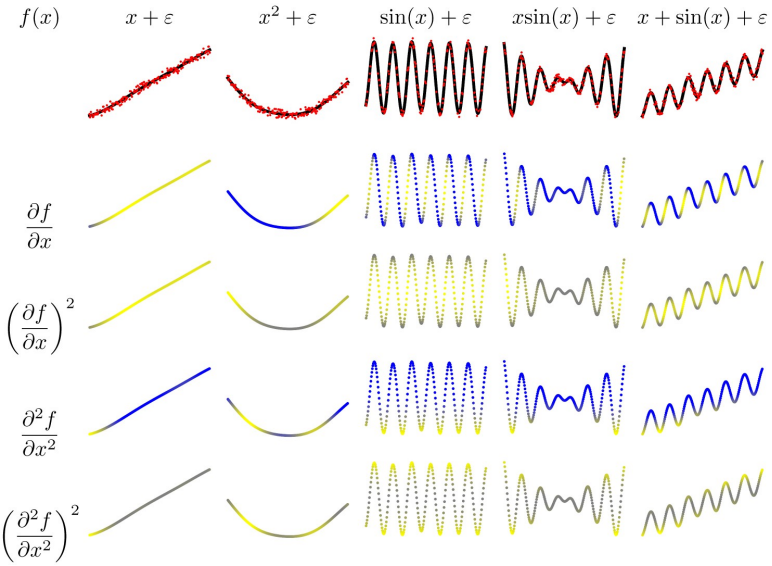


Fig 1. Different examples of functions, derivatives and sensitivity maps. Original data (red), the GP predictive function (black), high derivative values are in yellow, close-to-zero derivative values are in gray and negative derivative values are in blue.

https://doi.org/10.1371/journal.pone.0235885.g001

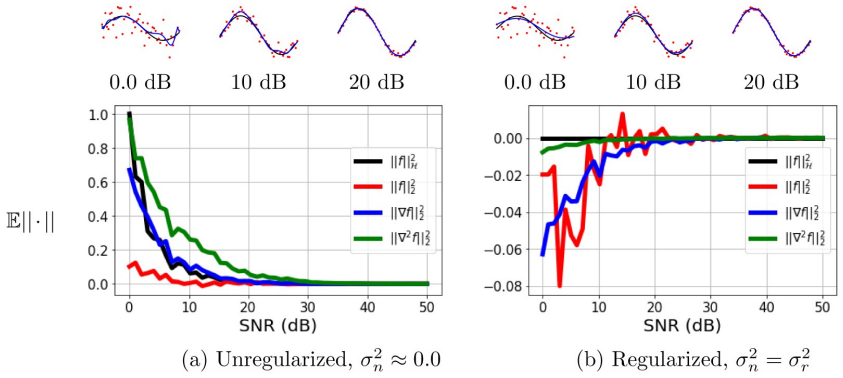
Gaussian noise (AWGN) $n \sim \mathcal{N}(0, \sigma_n^2)$ using a Kernel Ridge Regression (KRR) model with RBF kernel. Different amounts of noise power σ_n^2 was used resulting in different values of the signal to noise ratio (SNR), SNR = $10 \log(\sigma_y^2/\sigma_n^2)$, SNR$\in$[0, 50] dB. Two different settings were explored to analyze the impact of the standard regularizer, $\| f \|_{\mathcal{H}}^2$, and the derivatives in KRR modeling: (1) either using the optimal amount of regularization in (14), $\sigma_n^2 = \sigma_r^2$, or (2) assuming no regularization was needed, $\sigma_n^2 = 0$.

Four scenarios were explored in this experiment: $\| f \|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$, $\| f \|_2^2 = \boldsymbol{\alpha}^\top \mathbf{K}^\top \mathbf{K} \boldsymbol{\alpha}$, $\| \nabla f \|_2^2 = \boldsymbol{\alpha}^\top (\nabla \mathbf{K})^\top (\nabla \mathbf{K}) \boldsymbol{\alpha}$, and $\| \nabla^2 f \|_2^2 = \boldsymbol{\alpha}^\top (\nabla^2 \mathbf{K})^\top (\nabla^2 \mathbf{K}) \boldsymbol{\alpha}$, where $\mathbf{K}$ is a matrix with entries $[\mathbf{K}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (for definitions of gradients see Eqs (8) and (9)). The resulting SNR curves were then normalized in such a way that they are comparable. We explore two scenarios; the regularized and unregularized. Since the maximum SNR was subtracted from all norm values, in Fig 2a any norm greater than zero signifies the need to regularize more and in Fig 2b any norm less than zero signifies the need to regularize less.

Table 3. Summary of the formulation for each of the main kernel methods GPR (Gaussian Process Regression, section 3), SVM (Support Vector Machines, section 4), KDE (Kernel Density Estimation, section 5), HSIC (Hilbert-Schmidt Independence Criterion, section 6). The derivative formulation as well as some related analysis procedures in the literature as well as demonstrated in this paper.

Method	Function	Derivative	Analysis
GPR	$\mathbf{k}_*^\top \alpha$	$\partial \mathbf{k}_*^\top \alpha$	Sensitivity, Ranking, Regularization
SVM	$g(y \alpha \mathbf{k}_* + b)$	$(1 - g^2(\boldsymbol{x}_*)) \partial \mathbf{k}_*^\top y \alpha$	Sensitivity, Feature Ranking, Margin
KDE	$n^{-1} \boldsymbol{k}_* \mathbb{1}_n$	$\nabla \hat{p}(\boldsymbol{x}_*)^\top \mathbf{E}_r(\boldsymbol{x}_*)$	Principal Curves
HSIC	$n^{-2}\text{Tr}(\mathbf{KHLH})$	$2n^{-2} \mathbf{A}_i \partial_q \boldsymbol{k}(\mathbf{x}_i)$	Leverage, Feature/Point Relevance

https://doi.org/10.1371/journal.pone.0235885.t003



(a) Unregularized, $\sigma_n^2 \approx 0.0$ (b) Regularized, $\sigma_n^2 = \sigma_r^2$

Fig 2. Signal-to-Noise Ratio (SNR) versus the expected normalized value of different norms ($\mathbb{E}||\cdot||$) to act as regularizers. A unregularized (left) and an regularized (right) Kernel Ridge Regression (KRR) model was fitted. The top row shows a few examples of these fitted KRR models with a different quantity of noise added. The red data points are the data with different noise levels, the true function is black and the fitted KRR model is in blue. The second row shows the norm for the different regularizers. All lines were normalized in such a way that they are comparable. The norm of the true signal (SNR = 50 dB) is subtracted from all points so any curve with values below zero require less regularization and any points above zero require more regularization.

https://doi.org/10.1371/journal.pone.0235885.g002

Fig 2 shows the effect of the noise on the norm for different regularization terms. All four regularization functions give the user information about how noisy the signal is for the unregularized case and the regularized case. The graph for the regularized case has the norms of the functions below zero, except for the $\| f \|_{\mathcal{H}}^2$, when the SNR is extremely low. Since the norm of the functions are increasing as one increases the SNR, this says that there needs to be less regularization. The $\| f \|_{\mathcal{H}}^2$ has a straight line because the 'optimal' parameter for using the norm of the weights for regularization has already been chosen. However, the norm of the first and second derivative still give us information that the problem needs to be regularized less. So both cases showcase the functionality of the first and second derivative as viable regularizers.

4 Kernel classification

4.1 Support vector machine classification

The first effective and influential kernel method introduced was the Support Vector Machine (SVM) [1, 41–43] classifier. Researchers and practitioners have used it to solve problems in speech recognition [44], computer vision and image processing [45–47], or channel equalization [48]. The binary SVM classification algorithm minimizes a weighted sum of a loss and a regularizer

$$\sum_{i=1}^{n} \mathbf{V}(y_i, f(\boldsymbol{x}_i)) + \lambda \| f \|_{\mathcal{H}_k}^2,$$

where the cost function is called the 'hinge loss' and is defined as $\mathbf{V}(y_i, f(\boldsymbol{x}_i)) = \max(0, 1 - y_i \hat{f}(\boldsymbol{x}_i))$, $\boldsymbol{y}_i \in \{-1, +1\}$, $f \in \mathcal{H}_k$ and $\mathcal{H}_k$ is the RKHS of functions generated by the kernel k, and λ is a parameter that trades off accuracy for smoothness. The norm $\| f \|_{\mathcal{H}_K}$ is generally interpreted as a roughness penalty, and can be expressed as a

function of kernels, $\| f \|_{\mathcal{H}_K} = f^\top K f$. The decision function for any test point $\boldsymbol{x}_*$ is given by

$$\hat{y}_* = g(f(\boldsymbol{x})) = \operatorname{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i k(\boldsymbol{x}_*, \boldsymbol{x}_i) + b\right), \tag{16}$$

where α_i are Lagrange multipliers obtained from solving a quadratic programming (QP) problem, being the *support vectors* (SVs) of those training samples $\boldsymbol{x}_i$ with non-zero Lagrange multipliers $\alpha_i \neq 0$ [1]. See [49] for more details on the formulation and more practical examples.

4.2 Function derivatives and margin

The SVM decision function in (16) uses a mask function $g(x) = \operatorname{sgn}(\cdot)$ to decide between the two classes, which is inherited from the hinge loss used. Since the $\operatorname{sgn}(\cdot)$ function is not differentiable at 0 and for the sake of analytic tractability we replaced it with the hyperbolic tangent, $g(\cdot) = \tanh(\cdot)$. Now one can simply compute the derivative of the model by applying the chain rule:

$$\frac{\partial g(\boldsymbol{x}_*)}{\partial x_*^j} = \frac{\partial g(\boldsymbol{x}_*)}{\partial f(\boldsymbol{x}_*)} \frac{\partial f(\boldsymbol{x}_*)}{\partial x_*^j} = (1 - g^2(\boldsymbol{x})) \frac{\partial f(\boldsymbol{x}_*)}{\partial x_*^j} \tag{17}$$

where the leftmost term in the product can be seen as a mask function on top of the derivative of the regression function and allows us to study the model in terms of decision and estimation separately. See Table 3 for a comparison to other kernel methods derivatives.

Three datasets were used to illustrate the effect of the derivative in the SVM classifier. We used a SVM with RBF kernel in all cases, and hyperparameters were tuned by 3-fold cross-validation and the results are displayed in Fig 3. The mask function only focuses on regions along the decision boundary. However the derivative of the kernel function displays a few regions along the decision boundary along with other regions outside of the decision boundary. The composite of the derivative of the masking function and kernel function showcases a combination of the two components: the high derivative regions along the decision boundary. The two half moons and two circles examples have a clear decision boundary and the derivative of the composite function is able to capture this. However, the two ellipsoid example is less clear as the decision boundary passes through two overlapping classes. This is related to the density within the margin as the regions with less samples have a smaller slope and the regions with more samples have a higher slope, which results in wider and thinner margin, respectively. This fact could be used to define more efficient sampling procedures.

5 Kernel density estimation

The problem of density estimation is difficult in machine learning and statistics and it has been widely studied via kernels [50–52]. Kernel density estimation (KDE) is a classical non-parametric method for estimating a probability density function (pdf) [53]. In KDE, the choice of the kernel function is key to properly approximating the underlying pdf from a finite number of samples. The KDE kernel must be a non-negative function that integrates to one (i.e. a proper pdf), yet does not need to be positive semi-definite (PSD). KDE is versatile in that sense. However, if the kernel is PSD, there are close relations between density estimation and RKHS learning via the kernel eigendecomposition. Many KDE kernels are PSD, and some well-known examples include the Gaussian kernel, the Student kernel and the Laplacian kernel [54] functions.

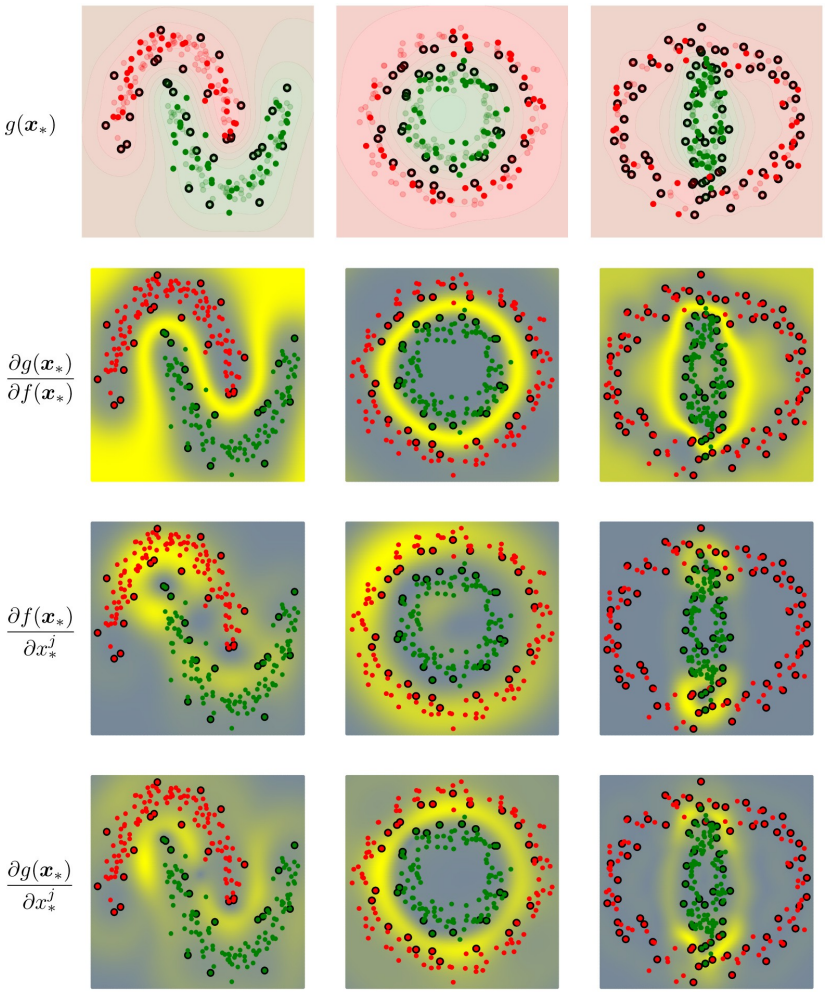


Fig 3. Visualizing three examples of sensitivity maps in SVM classification. The top row shows a figure has red and green points to showcase the classes, black points showing the support vectors chosen by the SVM classifier, and a contour map showcasing the same color scheme for the decision function. In the subsequent plots, we plot the sensitivity measures where the high derivative values are in yellow and negative derivative values are in gray. The leftmost column showcases which derivative is plotted.

https://doi.org/10.1371/journal.pone.0235885.g003

5.1 Density estimation with kernels

For the Parzen window expression, KDE defines the pdf as a sum of kernel functions defined on the training samples,

$$\hat{p}(\boldsymbol{x}_*) = \frac{1}{n}\sum_{i=1}^{n} k(\boldsymbol{x}_*, \boldsymbol{x}_i) = \frac{1}{n}\boldsymbol{k}_* \mathbb{1}_n, \tag{18}$$

where $\boldsymbol{k}_*$ is the vector of kernel evaluations between the point of interest $\boldsymbol{x}_*$, and all training samples (see section 3.1). KDE kernel functions have to be non-negative and integrate to one to ensure that $\hat{p}$ is a valid pdf. When a point-dependent weighting, β_i, is employed, then the above expression can be modified as $\hat{p}(\boldsymbol{x}_*) = \sum_{i=1}^{n} \beta_i k(\boldsymbol{x}_*, \boldsymbol{x}_i)$, where the β_i have to be positive and sum to one, i.e. $\beta_i \geq 0$ and $\sum_{i=1}^{n} \beta_i = 1$. In [51] a solution to find a suitable $\boldsymbol{\beta}$ vector based on kernel principal components analysis was proposed. If the decomposition of the un-centered kernel matrix follows the form $\mathbf{K} = \mathbf{E}\mathbf{D}\mathbf{E}^\top$, where $\mathbf{E}$ is orthonormal and $\mathbf{D}$ is a diagonal matrix, then the kernel-based density estimation can be expressed as

$$\hat{p}(\boldsymbol{x}_*) = \boldsymbol{k}_* \mathbf{E}_r \mathbf{E}_r^\top \mathbb{1}_n, \tag{19}$$

where $\mathbf{E}_r$ is the reduced version of $\mathbf{E}$ by keeping $r < n$ top eigenvectors. If we keep all the dimensions, i.e. $r = n$ the solution reduces to (18). By reducing the number of components we restrict the capacity of the density estimator and hence obtain a smoother approximation of the pdf as r reduces.

The retained kernel components should be selected by keeping the dimensions that maximize a sensible pdf characteristic, e.g. the variance. However, other criteria can be used to select the retained components. For instance, the kernel entropy component analysis (KECA) method uses the *information potential* as criterion to select the components from the eigenvector decomposition [8]. In this case, the decomposition method is already optimized to maximize the variance, therefore the solution will be sub-optimal. A more accurate way of finding a decomposition was presented in [55] where the features are directly optimized to maximize the amount of retained information. This method was named optimized KECA (OKECA), and showed excellent performance using very few extracted components.

The relevant aspect for this paper is that, by doing $\boldsymbol{\alpha} = \mathbf{E}_r \mathbf{E}_r^\top \mathbb{1}_n$, Eqs (18) and (19) can be cast in the general framework of kernel methods we proposed in Eq (4). Through this equality the derivatives and the second derivatives (and therefore the Hessian) can be obtained in a straightforward manner using Eqs (5) and (6). This information can be used for different problems, such as computing the Fisher's information matrix, optimizing vector quantization systems, or the example in the following section where we use them to find the points that belong to the principal curve of the distribution.

5.2 Derivatives and principal curves

This example illustrates the use of kernel derivatives in the KDE framework. In particular, we use the gradient and the Hessian of the pdf, to find points that belong to the *principal curve* along the data manifold [56]. A principal curve is defined as the curve that passes through the middle of the data. How to find this curve in practice is an important problem since multiple data description methods are based on drawing principal curves [30, 57–60]. In [30], they characterize the principal curve as the set of points that belong to the ridge of the density function. These points can be determined by using the gradient and the Hessian of the pdf: a point $\boldsymbol{x}_*$ is an element of the d-dimensional principal curve iff the inner product of the gradient, $\nabla\hat{p}(\boldsymbol{x}_*)$, and at least r eigenvectors of the Hessian, $\mathbf{H}(\boldsymbol{x}_*)$, is zero:

$$\nabla\hat{p}(\boldsymbol{x}_*)^\top \mathbf{E}_r(\boldsymbol{x}_*) = \mathbf{0}, \tag{20}$$

where $\mathbf{E}_r(\boldsymbol{x}_*)$ are the top r eigenvectors of the matrix $\mathbf{H}(\boldsymbol{x}_*)$. Note that applying this definition using our framework is straightforward as we can use the KDE to describe the probability density function, and Eqs (5) and (6), as well as formulas in Table 1, to find the gradient and the Hessian of the defined pdf with respect to the points. See Table 3 for a comparison to other kernel methods derivatives.

In Fig 4, we show an illustrative example of this application in three different toy datasets. The pdf can be obtained from the data points by using the OKECA method and the derivative lines describe the direction to which the density changes the most. The last row shows the points of the dataset with smaller dot products between the gradient and the last eigenvector of the Hessian, see Eq (20). Note that these points belong to the ridge of the distribution, and thus to the principal curve.

6 Kernel dependence estimation

6.1 Dependence estimation with kernel methods

Measuring dependencies and nonlinear associations between random variables is an active field of research. The kernel-based dependence estimation defines a covariance and cross-covariance operators in RKHS, and the subsequent statistics from these operators allows one to measure dependence between functions therein.

Let us consider two spacess $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ and $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$, which we jointly sample observation pairs $(\boldsymbol{x}, \boldsymbol{y})$ from distribution $\mathbb{P}_{\boldsymbol{xy}}$. The covariance matrix is $\mathcal{C}_{\boldsymbol{xy}} = \mathbb{E}_{\boldsymbol{xy}}(\boldsymbol{x}\boldsymbol{y}^\top) - \mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x})\mathbb{E}_{\boldsymbol{y}}(\boldsymbol{y}^\top)$, where $\mathbb{E}_{\boldsymbol{xy}}$ is the expectation with respect to $\mathbb{P}_{\boldsymbol{xy}}$, and $\mathbb{E}_{\boldsymbol{x}}$. A statistic that summarizes the content of the covariance matrix is its Hilbert-Schmidt norm. This quantity is zero if and only if there exists no second order dependence between $\boldsymbol{x}$ and $\boldsymbol{y}$.

The nonlinear extension of the notion of covariance was proposed in [13] to account for higher order statistics. Essentially, let us define a (possibly non-linear) mapping $\phi : \mathcal{X} \to \mathcal{F}$ such that the inner product between features is given by a PSD kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$. The feature space $\mathcal{F}$ has the structure of a RKHS. Similarly, we define $\psi : \mathcal{Y} \to \mathcal{G}$ with associated kernel function $l(\boldsymbol{y}, \boldsymbol{y}')$. Then, it is possible to define a cross-covariance operator between these feature maps, and to compute the squared norm of the cross-covariance operator, $\| \mathcal{C}_{\boldsymbol{xy}} \|^2_{\mathrm{HS}}$, which is called the Hilbert-Schmidt Independence Criterion (HSIC) and can be expressed in terms of kernels [61, 62]. Given a sample dataset $\mathcal{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ of size n drawn from $\mathbb{P}_{\boldsymbol{xy}}$, an empirical estimator of HSIC is [13]:

$$\mathrm{HSIC}(\mathcal{F}, \mathcal{G}, \mathbb{P}_{\boldsymbol{xy}}) = \frac{1}{n^2}\mathrm{Tr}(\mathbf{KHLH}) = \frac{1}{n^2}\mathrm{Tr}(\mathbf{HKHL}), \tag{21}$$

where $\mathrm{Tr}(\cdot)$ is the trace operation, $\mathbf{K}$, $\mathbf{L}$ are the kernel matrices for the input random variables $\boldsymbol{x}$ and $\mathbf{y}$ (i.e. $[\mathbf{K}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$), respectively, and $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^\top$ centers the data in the feature spaces $\mathcal{F}$ and $\mathcal{G}$, respectively. HSIC has demonstrated its capability to detect dependence between random variables but, as for any kernel method, the learned relations are hidden behind the kernel feature mapping. To address this issue, we consider the derivatives of HSIC.

6.2 Derivatives of HSIC

HSIC empirical estimate is parameterized as a function of two random variables, so the function derivatives given in section 2 are not directly applicable. Since HSIC is a symmetric measure, the solution for the derivative of HSIC wrt x_i^j will have the same form as the derivative wrt y_i^j. For convenience, we can group all terms that do not explicitly depend of $\mathbf{X}$ as $\mathbf{A} = \mathbf{HLH}$, which allows us expressing (21) simply as:

$$\mathrm{HSIC} := \frac{1}{n^2}\mathrm{Tr}(\mathbf{KA}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{A}]_{ij}k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{22}$$

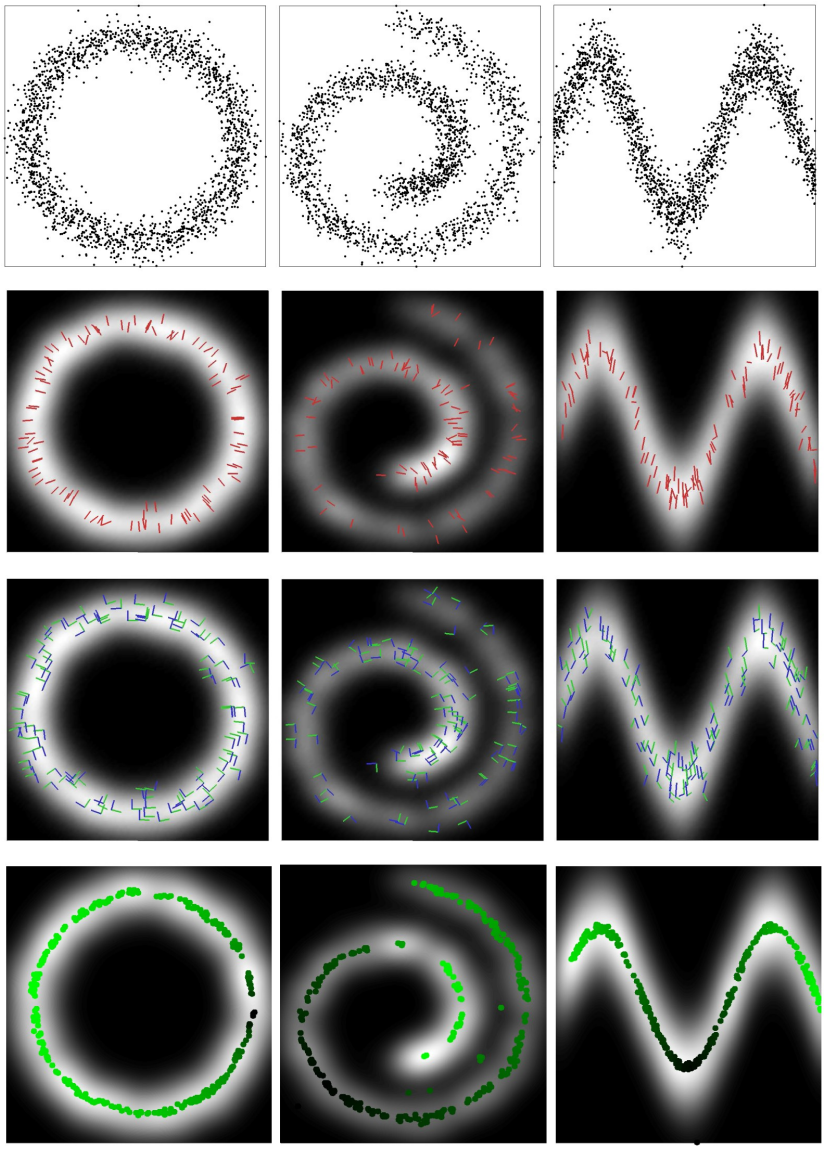


Fig 4. First row: Original data points. Second, third and fourth row: probability density in gray scale (brighter means denser). **Second row**: derivative direction of the pdf for some data points is represented using red lines. **Third row**: Hessian eigenvectors for some points represented with blue lines (first eigenvector) and green lines (second eigenvector). **Fourth row**: points on the ridge computed using the formula proposed in [30], different brightness of green has been computed using the Dijkstra distance over the curve dots (see text for details).

https://doi.org/10.1371/journal.pone.0235885.g004

Note that the core of the solution is the same as in the previous sections; a weighted combination of kernel similarities. However, now we need to derive both arguments of the kernel function k with respect to entry x_i^j that appears twice. By taking derivatives with regards to a particular dimension q of sample $\boldsymbol{x}_i$, i.e. x_i^q, and noting that the derivative of a kernel function is a symmetric operation, i.e. $\frac{\partial k(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\partial x_i^q} = \frac{\partial k(\boldsymbol{x}_j,\boldsymbol{x}_i)}{\partial x_i^q}$, one obtains

$$\frac{\partial \mathrm{HSIC}}{\partial x_i^q} = \frac{2}{n^2}\sum_{j=1}^{n}[\mathbf{A}]_{ij}\frac{\partial k(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\partial x_i^q} = \frac{2}{n^2}\mathbf{A}_i\partial_q \boldsymbol{k}(\mathbf{x}_i), \tag{23}$$

where $\mathbf{A}_i$ is the i-th row of the matrix $\mathbf{A}$. For the the RBF kernel we obtain [31]:

$$\frac{\partial \mathrm{HSIC}}{\partial x_i^q} = -\frac{2}{\sigma^2 n^2}\mathrm{Tr}(\mathbf{HLH}(\mathbf{K}\circ\mathbf{M}^q)), \tag{24}$$

where entries of matrix $\boldsymbol{M}^{\boldsymbol{q}}$ are $[\boldsymbol{M}^q]_{ij} = x_i^q - x_j^q$ $(1 \leq j \leq n)$, and zeros otherwise, and where the symbol $\circ$ is the Hadamard product between matrices.

Recently [63] extended the notion of leverage scores for the ridge regression problem. Leverage is a measure of how points with low density neighbours are enforcing the model for passing through them. By definition, the leverage (of a regressor) is the sensitivity of the predictive function w.r.t. the outputs. There is no definition of leverage in the case of HSIC as it is not a regression model but a dependence measure. However, HSIC could be interpreted in a similar way by fixing one of the variables and taking the derivative w.r.t. the other. By this interpretation, one can think of the HSIC sensitivity as a measure of how individual points are affecting the dependence measurement, i.e. how sensitive HSIC is to the perturbations for each particular point. This interpretation allows us to link the concepts of leverage and sensitivity in kernel dependence measures.

In this case, the derivatives of HSIC report information about the directions that impact the dependence estimate the most. This allows one to evaluate the measure as a *vector field* representation of two components. As in the previous kernel methods analyzed, the derivatives here are also analytic, just involving simple matrix multiplications and a trace operation. See Table 3 for a comparison to other kernel methods derivatives.

6.3 Visualizing kernel dependence measures

HSIC derivatives give information about the contribution of each point and feature to the dependence estimate. Fig 5 shows the directional derivative maps for three different bi-dimensional problems of variable association. We show the different components of the (sign-valued) vector field as well as its magnitude. In all problems, arrows indicate the strength of distortion to be applied to points (either in directions x, y, or jointly) such that the dependence is maximized. For the first example (top row), the map pushes the points into the 1-1 line and tries to collapse data into 2 different clusters along this line. In the second example (middle row), the distribution is a noisy ring: here the sensitivity map tries to collapse the data into clusters in order to maximize the dependence between the variables. In the last third experiment (bottom row), both variables are almost independent and the sensitivity map points towards some regions in the space where the dependence is maximized. In all cases, the S_x and S_y are orthogonal in direction and form a vector field whose intensity can be summarized in its norm $|S|$ (columns in the figure).

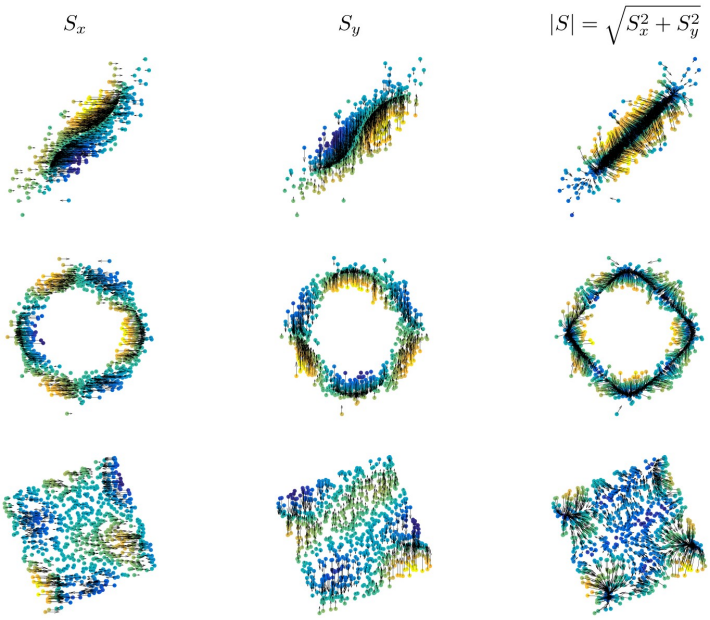


Fig 5. Visualizing the derivatives and the modulus of the directional derivative for HSIC in three toy examples.

https://doi.org/10.1371/journal.pone.0235885.g005

6.4 Unfolding and independization

We have seen that the derivatives of the HSIC function can be useful to learn about the data distribution and the variable associations. The derivatives of HSIC give information about the directions most affecting the dependence or independence measure.

Fig 6 shows an example of how the derivatives of the HSIC can be used to modify the data and achieve either maximum dependence or maximum independence. We embedded the derivatives in a simple gradient descent scheme, in which we move samples iteratively to maximize or minimize data dependence. Departing from a sinusoid, one can attain dependent or independent domains.

Note that HSIC can be understood as a maximum mean discrepancy (MMD) [64] between the joint probability measure of the involved variables and the product of their marginals, and MMD derivatives are very similar to those of HSIC provided here. The explicit use of the kernel derivatives would allow us to use gradient-descent approaches in methods that take advantage of HSIC or MMD, such as in algorithms for domain adaptation and generative modeling.

7 Analysis of spatio-temporal earth data

Kernel methods are widely applied in the Earth system sciences [5], where they have proven to be effective when dealing with low numbers of (potentially high dimensional) training samples. Data of this kind are characteristic for hyperspectral data, multidimensional sensor information, and different noise sources in the data. The most common applications in Earth system sciences are anomaly and target detection [65], the estimation of biogeochemical or biophysical parameters [66–68], dimensionality reduction [15, 69, 70], and the estimation of data interdependence [31]. However, so far multivariate spatio-temporal data problems have

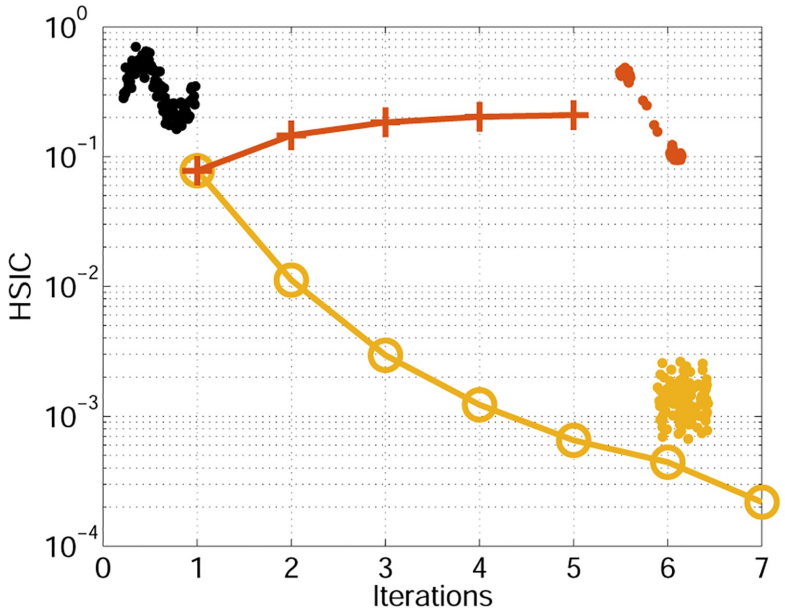


Fig 6. Modification of the input samples to maximize of minimize HSIC dependence between their dimensions (see text for details).

https://doi.org/10.1371/journal.pone.0235885.g006

received comparable little attention [71, 72], and in particular regarding the use of the derivatives of kernel methods [25, 26]. This is surprising, given the high-dimensional nature of most spatio-temporal dynamics in most sub-domains of the Earth system, e.g. land-surface dynamics, land-atmosphere interactions, ocean dynamics, etc. [73]. Hence, this section explores the added value of kernel derivatives for analyzing multivariate spatio-temporal Earth system data. We showcase applications considering the four studied problems of classification, regression, density estimation and dependence estimation. Please see github.com/IPL-UV/sakame for a working implementation of the algorithms as well as the subsequent ESDC experiments.

7.1 Spatio-temporal earth data

Today, data-driven research into Earth system dynamics has gained momentum and complements global modelling efforts. Much of Earth data is generated by a wide range of satellite sensors, upscaled products from *in-situ* observations, and model simulations with constantly improving spatial and temporal resolutions. The question is whether using kernel derivatives may help in (1) choosing the appropriate space and time scales to analyze phenomena, (2) visualize the most informative areas of interest, and (3) detect anomalies in spatio-temporal Earth data. We will work with products contained in the Earth System Data Lab (ESDL) [73]. The analysis-ready data-cube contains and harmonizes more than 40 variables relevant to monitor key processes of the terrestrial land-surface and atmosphere. The data streams contained in the ESDL are grouped in three data streams: land surface, atmospheric forcings and socio-economic data. Here we focus on three land-surface variables which exhibit nonlinear

relations in space and time. The following three variables; the gross primary productivity (GPP), root-zone soil moisture (SM), and land surface temperature (LST); are outlined below:

- GPP is the rate of fixation of carbon dioxide through the photosynthesis and one of the largest single flux in the global carbon cycle. However, the process is sensitive to climate variability. For instance, it has been shown that regional extreme events like droughts, heatwaves, and other types of disturbances may even influence the inter-annual variability of the globally integrated GPP [74]. Hence, it is key to understand the spatial and temporal dynamics of GPP at regional and global scales. Here, we consider the GPP FLUXCOM (http://www.fluxcom.org/) product, computed as described in [75, 76].
- **SM** plays a fundamental role for the environment and climate system, as it influences hydrological and agricultural processes, runoff generation and drought development processes, and land-atmospheric feedbacks [77] There are two products of soil moisture in our experiments. Standard SM products carry information limited to a few centimeters below the surface (±5 cm), and do not allow access to the whole zone from where water can be absorbed by roots. This is why we used root-zone soil moisture (RSM) [78–80] in the dependence estimation problem instead, a product from GLEAM that is a more sensitive variable to monitor water stress and droughts in vegetation.
- **LST** is an essential variable within the Earth climate system as it influences processes such as the exchange of energy and water between the land surface and atmosphere, and influences the rate and timing of plant growth. The LST product contained in the ESDL is the result of an ESA project called GlobTemperature, that developed a merged LST data set from thermal infrared (geostationary and polar orbiters) and passive microwave satellite data to provide best possible coverage.

The data is organized in 4-dimensional data cube $\mathbf{x}(u, v, t, k)$ involving (latitude, longitude) spatial coordinates (u, v), time sampling t, and the variable k. The data in ESDL contains a spatial resolution (high 0.083˚ resolution and coarser grid aggregation at 0.25˚) and a temporal resolution of 8 days spanning the years 2001-2011. In our experiments, we focus on the lower resolution products, during 2008-2010, and over Europe only. In the year 2010, a severe combination of spring and summer drought combined with a summer heat stress event affected large parts of Russia which can be observed in the three variables under study here [81], and we expect that also their interrelations must be affected. We use this well known event to provide a proof of concept for our suggestion approaches to interpret regressions, principal curves, and dependence estimation.

7.2 Sensitivity analysis in GP modelling

Studying time-varying processes with GPs is customary. Designing a GP becomes more complicated when dealing with spatio-temporal datasets. This can be cumbersome when the final goal is to understand and visualize spatial dependencies as well as to study the relevance of the features and the samples. Sensitivity analysis can be useful for either scenario. In this experiment, we study the impact of features in the GP modeling of the GPP and LST variables during 2010. To do so, we developed GP regression models trained to predict a pixel from their neighbourhood pixels. This is similar to geographically weighted regression [82] which can be used to model the local spatial relationships between these features and the outputs. From this framework, we can get sensitivity values for each of the contributing dimensions. We further split the data into subsets of spatial 'minicubes' which ranged in size from 2×2 until size 7×7. We use a GP model on a training subset of minicubes whereby the neighbours were used as

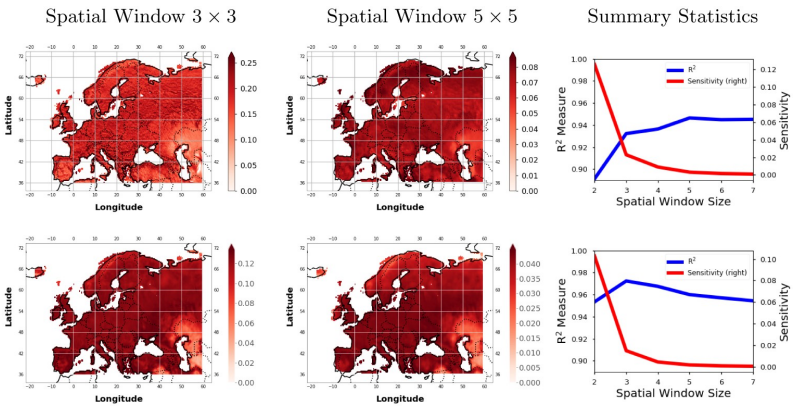


Fig 7. Visualizing the spatial maps for the senstivity of the Gaussian process (GP) regression model under different spatial sampling sizes for the Gross Primary Productivity (GPP) [top] and land surface temperature (LST) [bottom] for the summer of 2010 (Jun-Jul-Aug). The rightmost column shows the summary R^2 and Sensitivity for each spatial window size for the GP model.

https://doi.org/10.1371/journal.pone.0235885.g007

input dimensions to predict the center pixel for both GPP and LST. For metrics, we used the R^2-value to measure the goodness of fit between our model and the real data.

Fig 7 show how the sensitivity changes according to the mean prediction of the GPP and LST for two neighbourhood spatial window sizes (3 × 3 and 5 × 5). It also shows the spatial sensitivity maps for both settings and the R^2-value and the average sensitivity for each GP model. What's reassuring is that we see consistently low sensitivity values in areas (e.g. near the Black and Caspian Sea) for the GPP and LST regardless of the spatial window size as these are typically areas of low GPP and SM. For GPP, we see that sensitivities tend to become smoother as the neighbourhood size increases. These particular maps for GPP reach an R^2 value of 0.93 and 0.95 for each respective window size. Unlike the small differences in goodness of fit (+2% in R^2), the sensitivity curves show a wider variation and suggest that bigger windows are more appropriate to capture smoother areas; this is expected. Although we get a better model with a higher spatial window size, the sensitivity of neighbouring points become more dispersed over larger areas over Europe instead of just staying within small clusters. A similar pattern of dispersion of the sensitive points is observed for the LST maps w.r.t. the spatial window size. For LST, we notice that there is not a large difference in the R^2 as we increase the spatial window size. The most sensitive regions mostly stay the same but there is a small shift from the northern regions of Europe from more sensitive to less sensitive. So it's clear that the number of spatial-pixels used as input features would be different depending upon the input variable, e.g. one can use a higher neighbourhood size for LST because we get the same R^2 and similar sensitivity maps whereas the GPP could have a lower window size to ensure that we capture the local variability.

7.3 Classification of drought regions

Support vector machines (SVMs) is a very common classification method widely used in numerous applications in the field of machine learning in general and remote sensing in particular [5]. The derivatives of the SVM function, however, have not been used before to understand the model, nor linked to the concept of margin. The derivatives of SVMs can be broken

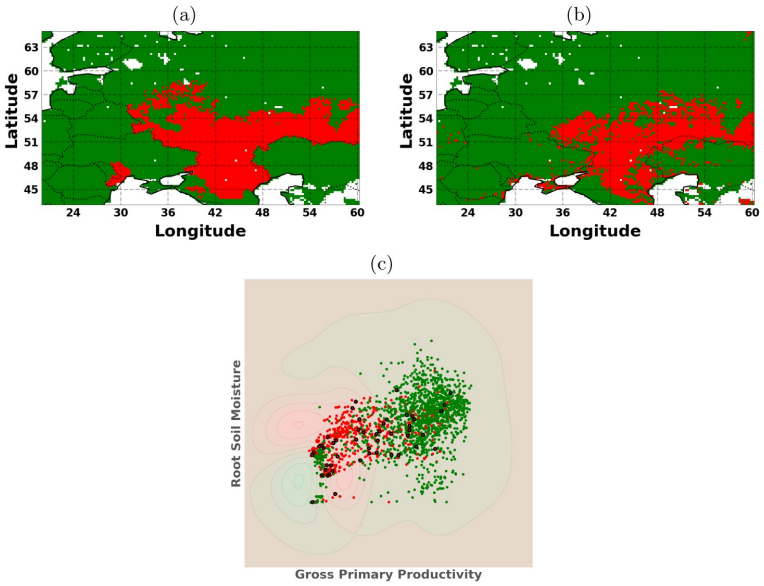


Fig 8. Visualizing the (a) labels, (b) predictions (b), and (c) the 2D representation space for the predictions. This is the classification problem of drought (red) versus no-drought (green) with the support vectors (black) for the SVM formulation (section 4).

https://doi.org/10.1371/journal.pone.0235885.g008

into two components via the product rule (section 4): 1) the derivative of the mask function and 2) the derivative of the kernel function. Typically one would use a sgn function as the mask but we used the *tanh* function to allow us to observe how the boundary or margin behaves w.r.t. the inputs.

In this experiment, we chose to study the relationship between gross primary productivity and root soil moisture during the year 2010 affected by a severe heatwave. There was a severe drought that occurred over this region for the entire summer of 2010 (June, July and August) [81]. Drought classification is an unsupervised problem and so there is a lot of debate about how to detect droughts within different scientific communities. We use a pre-defined drought mask of the countries affected by the 2010 heatwave found from the EM-DAT database [83] which reports all drought events which follow at least one of the criteria: 10 or more people dead, 100 or more people affected, declared state of emergency, or a call for international assistance. The region where the droughts are reported is just over the region of Eastern Europe, as shown in the binary classification maps in Fig 8. We chose this pre-defined drought area to simplify the problem which would allow us to see if we can indeed classify a drought region spatially and then look at the derivatives. We did a simple binary classification problem over the spatial coordinates using the two input variables (GPP and RM). We sampled only from the month of July at different time intervals within the month to make the samples more varied as the GPP and RM can still fluctuate within a monthly span. This is an unbalanced dataset as there are more non-drought regions than drought regions in the spatial subsample. While there are numerous advanced methods to deal with imbalanced datasets, we only used the standard SVM as that complexity is out of the scope for this experiment. The ESDC is very dense so we used 500 randomly selected points for the drought region and 1,000 randomly

Table 4. This table summarizes classification results for the drought and non-drought regions over Eastern Europe using the SVM (Support Vector Machines, section 4) formulation.

Class Label	Precision	Recall	F1-Score	Support
Non-Drought Regions	0.90	0.91	0.91	28590
Drought Regions	0.69	0.67	0.68	8268
Accuracy			0.86	36858

https://doi.org/10.1371/journal.pone.0235885.t004

selected points for the non-drought regions. The remaining points (∼ 3800) were considered for calculating test statistics, while the visualizations include all of the points for the dataset. We applied a standard cross-validated SVM classification algorithm with an RBF kernel function. For metrics, we used the standard precision, recall, F1-score and Support for the predictions of drought over non-drought. Table 4 shows the classification results compared to the labels of the trained SVM algorithm and Fig 8 shows the classification maps.

Fig 9 shows the sensitivity spatial maps as well as the 2D latent space for the outputs of the SVM classification model. We show the full derivative and the mask and kernel product components. The mask derivative has high sensitivity values for almost all regions where the decision function is unsure about the classification region. We see that the highest yellow regions are near the Caspian Sea which is also the area where there is a lot of overlap

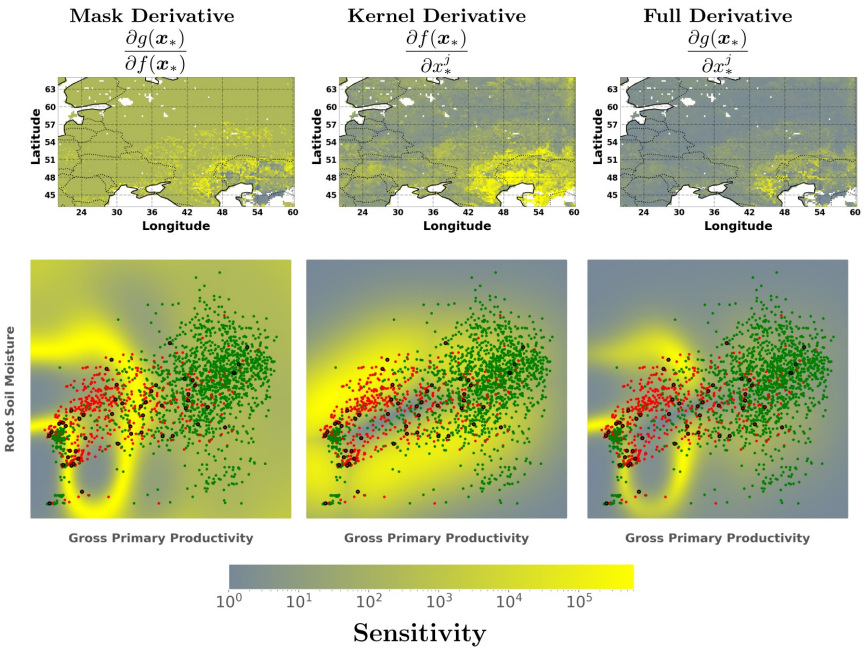


Fig 9. Visualizing the scatter plot of the drought (red) versus no-drought (green) and the support vectors (black) using the SVM section 4 classification algorithm. We also display the sensitivity of the full derivative and its components: the mask function (tanh) and the kernel function ($\partial \mathbf{k}$· αy) based on the predictive mean of the SVM classification results.

https://doi.org/10.1371/journal.pone.0235885.g009

between the classes. Recall that the mask used is based on the reported regions and not the actual GPP or SM values. So naturally the SVM algorithm is probably picking up the inconsistencies with the data given. Nevertheless, the kernel derivative indicates where there are regions of little data and in regions where there is significant overlap. Ultimately, the combination of the products represents a good balance between lack of data and the width of the margin between the two classes.

7.4 Principal curves of the ESDC

In this experiment we analyze GPP spatial-temporal patterns for different seasons for the year 2010 using the *principal curve* (PC) framework in Section 4. Each sample consists of a vector with the variable value for a particular location and all the time dimensions in the season: *(05-Jan to 05-May)*, *(21-May to 08-Aug)*, and *(17-Aug to 31-Dec)* For each season we have around 28, 000 samples of size $1 \times T$. Fig 10 shows the results. For each data set we plot the mean GPP value of the season in each point. The location of the points that belong to the PC are plotted in green using the Dijkstra distance inside the curve (as in the toy examples in Fig 4). The points belonging to the PC can be interpreted as the landmarks of the whole dataset, similar to a centroid of a cluster. But in this example, they refer to the points on the probability ridge of the data manifold (i.e. similar to the points closer to the first eigenvector in PCA). These points could be used for multiple purposes, e.g. as a summary to analyze the behaviour of the whole manifold or used for a temporal analysis of their evolution. One one hand, the location of the points is quite independent of the mean values, so they give different, alternative information. On the other hand, the location depends on the time of the year represented.

Most of the GPP 'representative' points are scattered around the manifold which depends on the season. For instance during the colder season (Jan-May), the dots are concentrated in the middle and low latitudes. During this period, the dots in northern Germany have a similar temperature and GPP than in the North-West part of Europe. Therefore, there is no need to add extra landmarks in these regions. Points in Morocco represent the warmer part of the manifold and Balcans area and Turkey represent the central part of the manifold. During the warmest period (May-Aug) the distribution of the dots follow an opposite direction, Southern regions are weighted less while Northern regions have more representation. In the case of mild temperatures (Aug-Dec), more landmarks in different regions are needed.

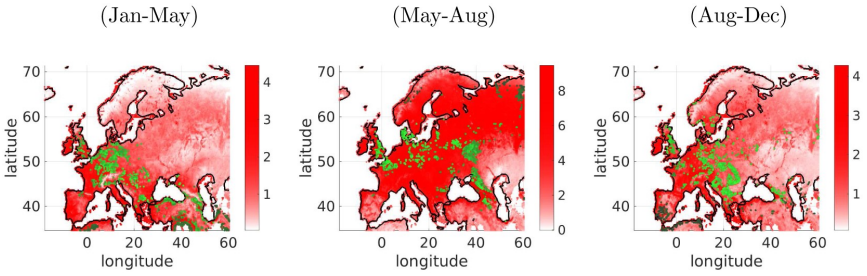


Fig 10. Principal curves on the ESDC. Each figure represents the results for GPP at different time periods during the 2010. In each image the mean value of the variable for each location is shown in colormap (minimum blue, maximum red), and the points that belong to the principal curves are represented in green. Different brightness of green has been computed using the Dijkstra distance over the curve dots.

https://doi.org/10.1371/journal.pone.0235885.g010

7.5 Sensitivity analysis of kernel dependence measures

HSIC is a dependence measure which can show differences in the higher-order relations between random variables. The derivatives of HSIC w.r.t. the input features are related to the change of the dependence measure which summarizes the relevance of the input features in the dependence. Therefore, these derivative maps can be related to the *sensitivity* of the inputs.

In this experiment, we chose to study the relation between GPP and RM for Europe and Russia during the years 2008, 2009 and 2010. We apply the HSIC with a linear kernel and compute the sensitivity maps, which is an estimation of how much the dependency criterion changes. We take spatial segments of GPP and RM at each time stamp, T and compute the HSIC value for each T independently for Russia and Europe. We also computed the derivative of HSIC for the same T time stamps independently for Russian and Europe. We computed the modulus to summarize the impact of each dimension to act as a proxy for the total average sensitivity. The final step involved computing the expectation between the modulus of the derivative of HSIC between Russian and Europe. Europe acts as a proxy stable environment and Russia is the one we would like to compare to. We estimated the expected value for three time periods (before: 05-Jan, 20-May; during: 28-May, 01-Sep; after: 09-Sep, 30-Dec) for each year individually. Then we compared each of the values to see how the expectation changes between Europe and Russia for each period across the years. The expected value of the HSIC derivatives summarize the change of association between variables differently than the HSIC measure itself.

The experiment focuses on studying the coupling/association between RM and GPP during the Russian drought in 2010. The HSIC algorithm captures an increased difference in dependencies of GPP and RM for Russia relative to Europe in 2010 if we compare this relationship to the years 2008 and 2009, see Fig 11a. However, HSIC only captures instantaneous instances of dependencies and not how fast these changes occur. The derivatives of HSIC (Fig 11b) allow us to quantify and capture when these changes actually occur. The gradients of HSIC do not show obvious differences in magnitude or shape across years between Russia and Europe. By taking the expected value of specific time periods of interest (before-during-after drought), we can highlight the contrast in the dependency trends between different periods with respect to their previous years, both in terms of HSIC and HSIC derivatives. We observe in Fig 11c, a change the mean value of the difference in the derivative of HSIC in Fig 11d which reveals a noticeable change in the trend for the springtime and summertime of 2010 compared to 2008 and 2009.

8 Conclusions

The use of Kernel methods is very popular in pattern analysis and machine learning and have been widely adopted because of their performance in many applications. However, they are still considered black-box models as the feature map is not directly accessible and predictions are difficult to interpret. In this note, we took a modest step back to understand different kernel methods by exploiting partial derivatives of the learned function with respect to the inputs.

To recap, we have provided intuitive explanations for derivatives of each kernel method through illustrative toy examples, and also highlighted the links between each of the formulations with concise expressions to showcase the similarities. We show that 1) the derivatives of kernel regression models (such as GPs) allows one to do sensitivity analysis to find relevant input features, 2) the derivatives of kernel classification models (such as SVMs) also allows one to do sensitivity analysis and visualize the margin, 3) the derivatives of kernel density estimators (KDE) allows one to describe the ridge of the estimated multivariate densities, 4) the derivatives of kernel dependence measures (such as HSIC) allows one to

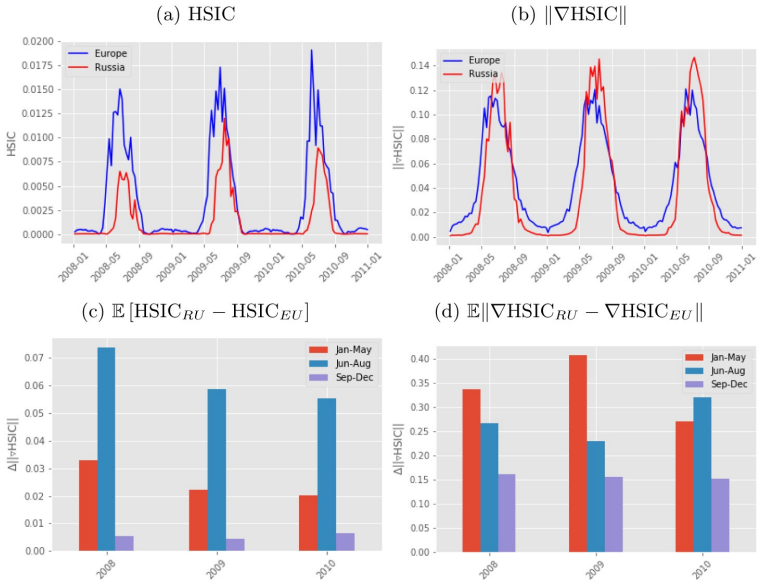


Fig 11. Each figure represents different summaries of how HSIC can be used to capture the differences in dependencies between Europe and Russia for GPP and RSM. (a) shows the HSIC value for Europe and Russia at each time stamp, (b) shows the derivative of HSIC for Europe and Russia at each time stamp, and the mean value for the difference in the (c) HSIC between Europe and Russia for different periods (Jan-May, Jun-Aug, and Sept-Dec), and (d) in the derivative of HSIC for the same periods.

https://doi.org/10.1371/journal.pone.0235885.g011

visualize the magnitude and change of direction in the dependencies between two multivariate variables. We have also given proof-of-concept examples of how they can be used in challenging applications with spatial-temporal Earth datasets. In particular, 1) we show that we can express the spatial-temporal relationships as inputs to regression algorithms and evaluate their relevance for prediction of essential climate variables, 2) we show that we can assess the margin for classification models in drought detection, as a way to identify the most sensitive points/regions for detection, 3) we show that the ridges can be used as indicators of potential regions of interest due to their location in the PDF, which could be related to anomalies, and 4) we show that we can detect changes in dependence between two events during an extreme heatwave event.

A Higher order derivatives of kernel functions

It can be shown that the m-th derivative of some kernel functions can be computed recursively using Faà di Bruno's identity [40] for the multivariate case:

$$\frac{\partial^{(m)}}{\partial x^{(m)}} f(g(\boldsymbol{x})) = \sum \frac{m!}{t_1!\, 1!^{t_1}\, t_2!\, 2!^{t_2} \cdots t_m!\, m!^{t_m}} \cdot \frac{\partial^{(t_1+\cdots+t_m)} f}{\partial g(\boldsymbol{x})} \cdot \prod_{i=1}^{n} \left(\frac{\partial g^{(i)}}{\partial x^{(i)}} \right)^{m_j},$$

where the sum is over all m-tuples $(t_1, \ldots, t_m) \in \mathbb{N}^m$ and $\sum_{j=1}^{m} j\, t_j = m$. It is also useful the

expression for mixed derivatives:

$$\frac{\partial^{(m)}}{\partial x^1 \cdots \partial x^m} f(g(\boldsymbol{x})) = \sum_{\pi \in \Pi} \frac{\partial^{|\pi|} f}{\partial g(\boldsymbol{x})} \cdot \prod_{B \in \pi} \frac{\partial g^{|B|}}{\prod_{j \in B} \partial x^{|B|}},$$

where Π is the ensemble all the partitions sets in $1 \ldots m$, π is a particular partition set, $B \in \pi$ runs over the blocks of the partition set π, and $|\pi|$ is the cardinality of π.

For the RBF kernel we can identify $f = \exp(\cdot)$ and $g = -\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2$. The derivatives for the $f(g(\boldsymbol{x}))$ are always the same $\partial^m f / \partial g(\boldsymbol{x})^m = f(g(\boldsymbol{x})) = \exp(g(\boldsymbol{x}))$, and the derivatives for the $g(\boldsymbol{x})$ are: $\partial g / \partial x^j = -2\gamma(x^j - y^j)$, $\partial^2 g / \partial x^{j2} = -2\gamma$, $\partial^m g / \partial x^{jm} = 0$, for $m \geq 3$, and $\frac{\partial^m g}{\partial x^1 \ldots \partial x^m} = 0$.

Applying the previous formula for $m = 1$ the first derivative is:

$$\begin{aligned} \frac{\partial}{\partial x^j} f(g(\boldsymbol{x})) &= \frac{\partial f}{\partial g(\boldsymbol{x})} \frac{\partial g}{\partial x^j} \\ &= f(g(\boldsymbol{x}))(-2\gamma(x^j - y^j)) \\ &= -2\gamma(x^j - y^j) k(\boldsymbol{x}, \boldsymbol{y}). \end{aligned}$$

The second derivative is:

$$\begin{aligned} \frac{\partial^2}{\partial x^{j2}} f(g(\boldsymbol{x})) &= \frac{\partial f}{\partial g(\boldsymbol{x})} \frac{\partial^2 g}{\partial x^{j2}} + \frac{\partial^2 f}{\partial g(\boldsymbol{x})^2} \left(\frac{\partial g}{\partial x^j} \right)^2 \\ &= f(g(\boldsymbol{x}))(-2\gamma) + f(g(\boldsymbol{x}))(4\gamma^2 (x^j - y^j)^2) \\ &= 2\gamma(2\gamma(x^j - y^j)^2 - 1) k(\boldsymbol{x}, \boldsymbol{y}). \end{aligned}$$

The mixed derivative is:

$$\begin{aligned} \frac{\partial}{\partial x^j \partial x^i} f(g(\boldsymbol{x})) &= \frac{\partial f}{\partial g(\boldsymbol{x})} \frac{\partial^2 g}{\partial x^j \partial x^i} + \frac{\partial^2 f}{\partial g(\boldsymbol{x})^2} \left(\frac{\partial g}{\partial x^j} \right) \left(\frac{\partial g}{\partial x^i} \right) \\ &= f(g(\boldsymbol{x}))(0) + f(g(\boldsymbol{x}))(-2\gamma^2 (x^j - y^j))(-2\gamma^2 (x^i - y^i)) \\ &= 4\gamma^2 (x^j - y^j)(x^i - y^i) k(\boldsymbol{x}, \boldsymbol{y}). \end{aligned}$$

B Custom regression function

In this example we show the behaviour of the first and second derivatives for a multivariate input. A GP model is fitted over the dataset using the RBF kernel function. The experiment uses a custom linear multivariate function with two inputs, x_1 and x_2, as inputs:

$$y = a x_1 + b x_2, \tag{25}$$

where the coefficients a and b have varying values. Both $x_{1,2}$ were generated along the same range uniform distribution $\mathcal{U}([-20, 20])$ but there was a linear transformation $a = 5$, $b = 1$ from $([0, 20])$ and constant everywhere else, i.e. $a = b = 1$ from $([-20, 0])$.

The GP model smooths the piece-wise continuous function which results in some additional slopes than the original formulation. This is visible (see Fig 12) from the derivatives of the kernel model as the first derivative for the x_1 and x_2 components have positive values for the sensitivities of the slopes in the regions where a and b are equal to some constant, respectively. The second derivative for both x_1 and x_2 show the same effect except for curvature. This experiment successfully highlights the derivatives of the individual components as well as their combined sensitivity.

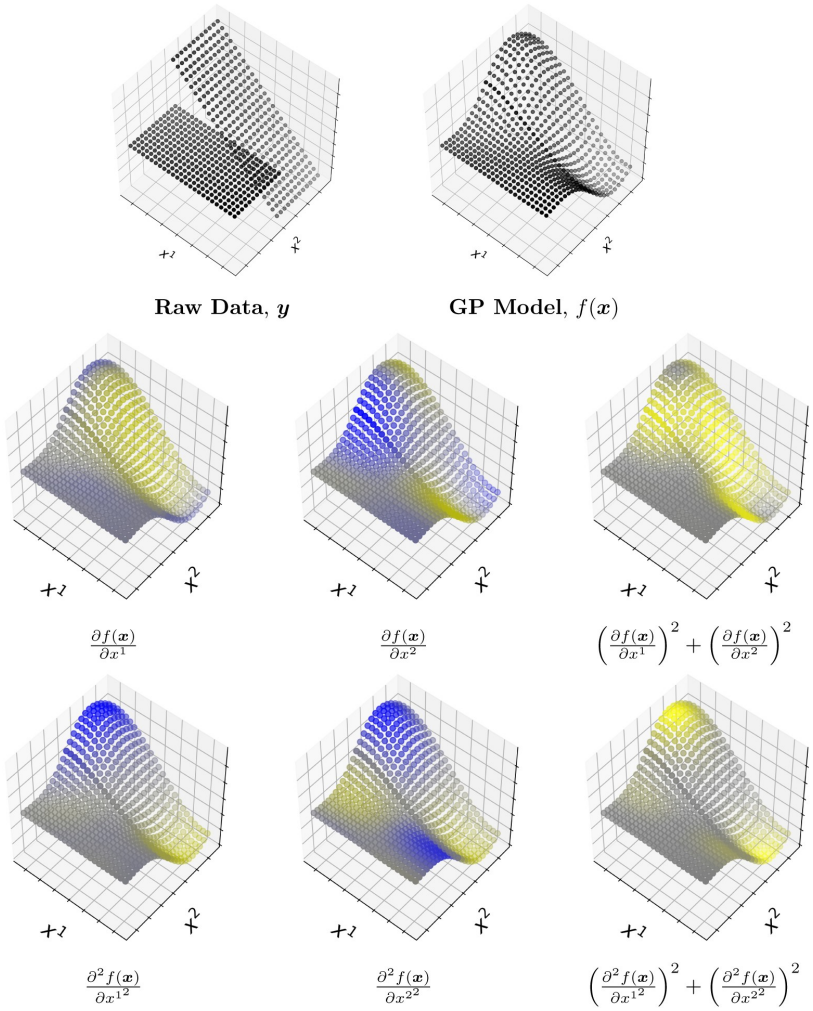


Fig 12. First row: The original toy data is displayed as well as the predicted GP model which presents a smoother curve. **Second row**: the first derivative in the x_1,x_2 direction and combined direction (the sensitivity) respectively. **Third row**: the second derivative in the x_1, x_2 direction and combined direction (the sensitivity) respectively. The yellow colored points represent the regions with positive values, the blue colored points represent the regions with negative values and the gray colored points represent the regions where the values are zero.

https://doi.org/10.1371/journal.pone.0235885.g012

Acknowledgments

J.E.J. thanks the European Space Agency (ESA) for support via the Early Adopter Call of the Earth System Data Lab project; M.D.M. thanks the ESA for the long-term support of this initiative.

Author Contributions

Conceptualization: J. Emmanuel Johnson, Valero Laparra, Adrián Pérez-Suay, Miguel D. Mahecha, Gustau Camps-Valls.

Data curation: J. Emmanuel Johnson, Miguel D. Mahecha.

Formal analysis: J. Emmanuel Johnson, Valero Laparra, Adrián Pérez-Suay, Gustau Camps-Valls.

Funding acquisition: Gustau Camps-Valls.

Investigation: Valero Laparra, Adrián Pérez-Suay, Gustau Camps-Valls.

Methodology: J. Emmanuel Johnson.

Resources: J. Emmanuel Johnson, Miguel D. Mahecha, Gustau Camps-Valls.

Software: J. Emmanuel Johnson, Adrián Pérez-Suay.

Supervision: Valero Laparra, Miguel D. Mahecha, Gustau Camps-Valls.

Validation: J. Emmanuel Johnson, Valero Laparra, Adrián Pérez-Suay, Gustau Camps-Valls.

Visualization: Valero Laparra.

References

1. Schölkopf B, Smola A. Learning with kernels-Support Vector Machines, Regularization, Optimization and Beyond. MIT Press. 2002;.
2. Shawe-Taylor J, Cristianini N. Kernel Methods for Pattern Analysis. Cambridge University Press; 2004.
3. Rojo-Álvarez JL, Martínez-Ramón M, Muñoz-Marí J, Camps-Valls G. Digital Signal Processing with Kernel Methods. UK: Wiley & Sons; 2017.
4. Lampert CH. Kernel Methods in Computer Vision. Foundations and Trends® in Computer Graphics and Vision. 2009; 4(3):193–285. https://doi.org/10.1561/0600000027
5. Camps-Valls G, Bruzzone L. Kernel methods for Remote Sensing Data Analysis. Camps-Valls G, Bruzzone L, editors. UK: Wiley & Sons; 2009. https://doi.org/10.1002/9780470748992
6. Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. Cambridge, MA: The MIT Press; 2006.
7. Smola AJ, Schölkopf B. A tutorial on support vector regression. Statistics and Computing. 2004; 14:199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88
8. Jenssen R. Kernel Entropy Component Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010; 31(9).
9. Schölkopf B, Smola A, Müller K. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation. 1998; 10(5).
10. Lai PL, Fyfe C. Kernel and non-linear Canonical Correlation Analysis. Intl Journal of Neural Systems. 2000; 10:365–377. https://doi.org/10.1142/S012906570000034X
11. Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing Hilbert spaces. Journal of Machine Learning Research. 2001; 2:97–123.
12. Gretton A, Herbrich R, Hyvärinen A. Kernel methods for measuring independence. Journal of Machine Learning Research. 2005; 6:2075–2129.
13. Gretton A, Bousquet O, Smola A, Schölkopf B. Measuring Statistical Dependence with Hilbert-Schmidt Norms. In: Jain S, Simon H, Tomita E, editors. Algorithmic Learning Theory. vol. 3734 of Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2005. p. 63–77.
14. Quadrianto N, Song L, Smola AJ. Kernelized Sorting. In: Koller D, Schuurmans D, Bengio Y, Bottou L, editors. Advances in Neural Information Processing Systems 21. Curran Associates, Inc.; 2009. p. 1289–1296.
15. Tuia D, Camps-Valls G. Kernel Manifold Alignment for Domain Adaptation. PLOS ONE. 2016 2; 11 (2):1–25. https://doi.org/10.1371/journal.pone.0148655

16. Martínez-Ramón M, Rojo-Álvarez JL, Camps-Valls G, Muñoz-Marí J, Navia-Vazquez A, Soria-Olivas E, et al. Support vector machines for nonlinear kernel ARMA system identification. IEEE Transactions on Neural Networks. 2006 Nov; 17(6):1617–1622. https://doi.org/10.1109/TNN.2006.879767

17. Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. New York: The MIT Press; 2006.

18. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y. SimpleMKL. Journal of Machine Learning Research. 2008 Nov; 9:2491–2521.

19. Burges C. Geometry and Invariance in Kernel Based Methods. Cambridge, USA: Eds. Schölkopf B., Burges C., Smola A., MIT Press; 1998.

20. Bakir G, Weston J, Schölkopf B. Learning to find Pre-images. NIPS; 2003. p. 449–456.

21. Kwok JT, Tsang IW. The Pre-Image Problem in Kernel Methods. IEEE Trans Neural Networks. 2004; 15(6):1517–1525. https://doi.org/10.1109/TNN.2004.837781 PMID: 15565778

22. Wahba G. Splines in Nonparametric Regression. Wiley & Sons, Ltd; 2006.

23. Kjems U, Hansen LK, Anderson J, Frutiger S, Muley S, Sidtis J, et al. The Quantitative Evaluation of Functional Neuroimaging Experiments: Mutual Information Learning Curves. NeuroImage. 2002; 15 (4):772–786. https://doi.org/10.1006/nimg.2001.1033 PMID: 11906219

24. Rasmussen PM, Madsen KH, Lund TE, Hansen LK. Visualization of nonlinear kernel models in neuroimaging by sensitivity maps. NeuroImage. 2011; 55(3):1120–1131. https://doi.org/10.1016/j.neuroimage.2010.12.035 PMID: 21168511

25. Camps-Valls G, Jung M, Ichii K, Papale D, Tramontana G, Bodesheim P, et al. Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International; 2015. p. 4416–4419.

26. Blix K, Camps-Valls G, Jenssen R. Sensitivity analysis of Gaussian processes for oceanic chlorophyll prediction. 2015 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2015, Milan, Italy, July 26-31, 2015. 2015;p. 996–999.

27. Mchutchon A, Rasmussen CE. Gaussian Process Training with Input Noise. In: Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira F, Weinberger KQ, editors. Advances in Neural Information Processing Systems 24. Curran Associates, Inc.; 2011. p. 1341–1349.

28. Johnson JE, Laparra V, Camps-Valls G. Accounting for Input Noise in Gaussian Process Parameter Retrieval. IEEE Geoscience and Remote Sensing Letters. 2020; 17(3):391–395. https://doi.org/10.1109/LGRS.2019.2921476

29. Martino L, Elvira V, Camps-Valls G. Group Importance Sampling for Particle Filtering and MCMC; 2017.

30. Ozertem U, Erdogmus D. Locally Defined Principal Curves and Surfaces. Journal of Machine Learning Research. 2011; 12:1249–1286.

31. Pérez-Suay A, Camps-Valls G. Sensitivity maps of the Hilbert–Schmidt independence criterion. Applied Soft Computing. 2017;.

32. Aizerman MA, Braverman EM, Rozoner LI. Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning. Automation and remote Control. 1964; 25:821–837.

33. Aronszajn N. Theory of reproducing kernels. Transactions of the American Mathematical Society. 1950 May; 68(3):337–404. https://doi.org/10.1090/S0002-9947-1950-0051437-7

34. Riesz F, Nagy BS. Functional Analysis. Frederick Ungar Publishing Co.; 1955.

35. Kimeldorf G, Wahba G. Some results on Tchebycheffian spline functions. Journal of Mathematical Analysis and Applications. 1971; 33(1):82–95. https://doi.org/10.1016/0022-247X(71)90184-3

36. Schölkopf B, Herbrich R, Smola AJ. A Generalized Representer Theorem. In: Helmbold D, Williamson B, editors. Computational Learning Theory. Berlin, Heidelberg: Springer Berlin Heidelberg; 2001. p. 416–426.

37. Gnecco G, Sanguineti M. Accuracy of suboptimal solutions to kernel principal component analysis. Computational Optimization and Applications. 2009 Mar; 42(2):265–287. https://doi.org/10.1007/s10589-007-9108-y

38. Cucker F, Smale S. On the mathematical foundations of learning. Bulletin of the American Mathematical Society. 2002; 39:1–49. https://doi.org/10.1090/S0273-0979-01-00923-5

39. Manton JH, Amblard PO. A Primer on Reproducing Kernel Hilbert Spaces. Foundations and Trends in Signal Processing. 2014; 8:1–126. https://doi.org/10.1561/2000000050

40. Arbogast LFA. Du calcul des derivations [microform] / par L.F.A. Arbogast. Levrault Strasbourg; 1800.

41. Boser BE, Guyon I, Vapnik VN. A training algorithm for optimal margin classifiers. In: Proc. COLT'92. PA.Pittsburgh, PA: Pittsburgh; 1992. p. 144–152.

42. Cortes C, Vapnik V. Support Vector Networks. Machine Learning. 1995; 20:273–97. https://doi.org/10.1023/A:1022627411411

43. Vapnik V. Statistical Learning Theory, Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons; 1998.
44. Xing H, Hansen J. Single Sideband Frequency Offset Estimation and Correction for Quality Enhancement and Speaker Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2017; 25(1):124–136. https://doi.org/10.1109/TASLP.2016.2623563
45. Cremers D, Kohlberger T, Schnörr C. Shape statistics in kernel space for variational image segmentation. Pattern Recognition. 2003; 36(9):1929—1943. Kernel and Subspace Methods for Computer Vision.
46. Kim KI, Franz MO, Scholkopf B. Iterative Kernel Principal Component Analysis for Image Modeling. IEEE Trans Pattern Anal Mach Intell. 2005 Sep; 27(9):1351–1366. https://doi.org/10.1109/TPAMI.2005.181 PMID: 16173181
47. Xu M, Jiang L, Sun X, Ye Z, Wang Z. Learning to Detect Video Saliency With HEVC Features. IEEE Transactions on Image Processing. 2017; 26(1):369–385. https://doi.org/10.1109/TIP.2016.2628583
48. Chen S, Gunn S, Harris CJ. Decision Feedback Equalizer Design Using Support Vector Machines. In: IEE Proc. Vision, Image and Signal Processing, Vol.147, No.3; 2000. p. 213–219.
49. Burges CJC. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery. 1998; 2(2):121–167. https://doi.org/10.1023/A:1009715923555
50. Silverman B. Density Estimation for Statistics and Data Analysis. London, England: Chapman and Hall; 1986.
51. Girolami M. Orthogonal series density estimation and the kernel eigenvalue problem. Neural Computation. 2002; 14(3):669–688. https://doi.org/10.1162/089976602317250942 PMID: 11860687
52. Duin RPW. On the Choice of Smoothing Parameters for Parzen Estimators of Probability Density Functions. Computers, IEEE Transactions on. 1976;C- 25(11):1175–1179. https://doi.org/10.1109/TC.1976.1674577
53. Parzen E. On Estimation of a Probability Density Function and Mode. The Annals of Mathematical Statistics. 1962; 33(3):1065–1076. https://doi.org/10.1214/aoms/1177704472
54. Kim J, Scott CD. Robust Kernel Density Estimation. J Mach Learn Res. 2012 Sep; 13(1):2529–2565.
55. Izquierdo-Verdiguier E, Laparra V, Jenssen R, Gómez-Chova L, Camps-Valls G. Optimized Kernel Entropy Components. IEEE Transactions on Neural Networks and Learning Systems. 2017 June; 28(6):1466–1472. https://doi.org/10.1109/TNNLS.2016.2530403 PMID: 26930695
56. Hastie T, Stuetzle W. Principal curves. Journal of the American Statistical Association. 1989; 84:502–516. https://doi.org/10.1080/01621459.1989.10478797
57. Laparra V, Malo J, Camps-Valls G. Dimensionality Reduction via Regression in Hyperspectral Imagery. IEEE Journal of Selected Topics in Signal Processing. 2015 Sept; 9(6):1026–1036. https://doi.org/10.1109/JSTSP.2015.2417833
58. Laparra V, Jiménez S, Tuia D, Camps-Valls G, Malo J. Principal polynomial analysis. International Journal of Neural Systems. 2014; 24(7).
59. Laparra V, Jiménez S, Camps-Valls G, Malo J. Nonlinearities and adaptation of color vision from sequential principal curves analysis. Neural Computation. 2012; 24(10):2751–2788. https://doi.org/10.1162/NECO_a_00342 PMID: 22845821
60. Sasaki H, Kanamori T, Sugiyama M. Estimating Density Ridges by Direct Estimation of Density-Derivative-Ratios. In: Singh A, Zhu J, editors. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. vol. 54 of Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR; 2017. p. 204–212.
61. Baker C. Joint measures and cross-covariance operators. Transactions of the American Mathematical Society. 1973; 186:273–289. https://doi.org/10.1090/S0002-9947-1973-0336795-3
62. Fukumizu K, Bach FR, Jordan MI. Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces. Journal of Machine Learning Research. 2004; 5:73–99.
63. Alaoui A, Mahoney MW. Fast Randomized Kernel Ridge Regression with Statistical Guarantees. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, editors. Advances in Neural Information Processing Systems 28. Curran Associates, Inc.; 2015. p. 775–783.
64. Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A. A Kernel Two-sample Test. J Mach Learn Res. 2012 Mar;13:723–773. Available from: http://dl.acm.org/citation.cfm?id=2188385.2188410.
65. Capobianco L, Garzelli A, Camps-Valls G. Target detection with semisupervised kernel orthogonal subspace projection. IEEE Transactions on Geoscience and Remote Sensing. 2009; 47(11):3822–3833. Available from: http://www.scopus.com/inward/record.url?eid=2-s2.0-70350676140&partnerID=40&md5=a4a3777adc4aaab2807f8efa27ca0e02. https://doi.org/10.1109/TGRS.2009.2020910

66. Verrelst J, Alonso L, Camps-Valls G, Delegido J, Moreno J. Retrieval of vegetation biophysical parameters using Gaussian process techniques. IEEE Transactions on Geoscience and Remote Sensing. 2012; 50(5 PART 2):1832–1843. Available from: http://www.scopus.com/inward/record.url?eid=2-s2.0-84860332507&partnerID=40&md5=8f89c24c1927827bf249a795b35098fc. https://doi.org/10.1109/TGRS.2011.2168962

67. Camps-Valls G, Verrelst J, Muñoz-Marí J, Laparra V, Mateo-Jimenez F, Gomez-Dans J. A Survey on Gaussian Processes for Earth Observation Data Analysis: A Comprehensive Investigation. IEEE Geoscience and Remote Sensing Magazine. 2016 June;(6). Available from: http://ieeexplore.ieee.org/document/7487896/.

68. Camps-Valls G, Sejdinovic D, Runge J, Reichstein M. A Perspective on Gaussian Processes for Earth Observation. National Science Review. 2019 Mar; 6(4):616–618. https://doi.org/10.1093/nsr/nwz028

69. Mahecha MD, Fürst LM, Gobron N, Lange H. Identifying multiple spatiotemporal patterns: A refined view on terrestrial photosynthetic activity. Pattern Recognition Letters. 2010; 31(14):2309–2317. https://doi.org/10.1016/j.patrec.2010.06.021

70. Gómez-Chova L, Jenssen R, Camps-Valls G. Kernel entropy component analysis for remote sensing image clustering. IEEE Geoscience and Remote Sensing Letters. 2012; 9(2):312–316. Available from: http://www.scopus.com/inward/record.url?eid=2-s2.0-84856976376&partnerID=40&md5=e5bb9e506490459dd504d3cae854bc6f. https://doi.org/10.1109/LGRS.2011.2167212

71. Bueso D, Piles M, Camps-Valls G. Nonlinear PCA for Spatio-Temporal Analysis of Earth Observation Data. IEEE Transactions on Geoscience and Remote Sensing. 2020;. https://doi.org/10.1109/TGRS.2020.2969813

72. Lin Y, Yu J, Cai J, Sneeuw N, Li F. Spatio-temporal analysis of wetland changes using a kernel extreme learning machine approach. Remote Sensing. 2018; 10(7):1129. https://doi.org/10.3390/rs10071129

73. Mahecha MD, Gans F, Brandt G, Christiansen R, Cornell SE, Fomferra N, et al. Earth system data cubes unravel global multivariate dynamics. Earth System Dynamics. 2020; 11(1):201–234. Available from: https://www.earth-syst-dynam.net/11/201/2020/. https://doi.org/10.5194/esd-11-201-2020

74. Zscheischler J, Mahecha MD, von Buttlar J, Harmeling S, Jung M, Rammig A, et al. Few extreme events dominate global interannual variability in gross primary production. Environmental Research Letters. 2014; 9:035001. https://doi.org/10.1088/1748-9326/9/3/035001

75. Tramontana G, Jung M, Schwalm CR, Ichii K, Camps-Valls G, Ráduly B, et al. Predicting carbon dioxide and energy fluxes across global FLUXNET sites with regression algorithms. Biogeosciences. 2016; 13(14):4291–4313. Available from: http://www.biogeosciences.net/13/4291/2016/. https://doi.org/10.5194/bg-13-4291-2016

76. Jung M, Schwalm C, Migliavacca M, Walther S, Camps-Valls G, Koirala S, et al. Scaling carbon fluxes from eddy covariance sites to globe: synthesis and evaluation of the FLUXCOM approach. Biogeosciences. 2020; 17(5):1343–1365. Available from: https://www.biogeosciences.net/17/1343/2020/. https://doi.org/10.5194/bg-17-1343-2020

77. Miralles DG, Gentine P, Seneviratne SI, Teuling AJ. Land–atmospheric feedbacks during droughts and heatwaves: state of the science and current challenges. Annals of the New York Academy of Sciences. 2019; 1436(1):19. https://doi.org/10.1111/nyas.13912 PMID: 29943456

78. Martens B, Miralles DG, Lievens H, van der Schalie R, de Jeu RAM, Fernández-Prieto D, et al. GLEAM v3: satellite-based land evaporation and root-zone soil moisture. Geoscientific Model Development. 2017; 10(5):1903–1925. Available from: https://www.geosci-model-dev.net/10/1903/2017/. https://doi.org/10.5194/gmd-10-1903-2017

79. Dorigo W, Wagner W, Albergel C, Albrecht F, Balsamo G, Brocca L, et al. ESA CCI Soil Moisture for improved Earth system understanding: State-of-the art and future directions. Remote Sensing of Environment. 2017; 203:185—215. Earth Observation of Essential Climate Variables. https://doi.org/10.1016/j.rse.2017.07.001

80. Liu YY, Dorigo WA, Parinussa RM, de Jeu RAM, Wagner W, McCabe MF, et al. Trend-preserving blending of passive and active microwave soil moisture retrievals. Remote Sensing of Environment. 2012; 123:280—297. https://doi.org/10.1016/j.rse.2012.03.014

81. Flach M, Sippel S, Gans F, Bastos A, Brenning A, Reichstein M, et al. Contrasting biosphere responses to hydrometeorological extremes: revisiting the 2010 western Russian heatwave. Biogeosciences. 2018; 15(20):6067–6085. Available from: https://www.biogeosciences.net/15/6067/2018/. https://doi.org/10.5194/bg-15-6067-2018

82. Charlton AB. Geographically Weighted Regression: The Analysis of Spatially Varying Relationships. Wiley & Sons, Ltd; 2002.

83. EM-DAT. EM-DAT: The International Disaster Database; 2008. Available at: http://www.emdat.be/Database/Trends/trends.html.

D.2. Paper II

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Accounting for Input Noise in Gaussian Process Parameter Retrieval

Juan Emmanuel Johnson, Valero Laparra, and Gustau Camps-Valls, *Fellow, IEEE*

***Abstract*— Gaussian processes (GPs) are a class of Kernel methods that have shown to be very useful in geoscience and remote sensing applications for parameter retrieval, model inversion, and emulation. They are widely used because they are simple, flexible, and provide accurate estimates. GPs are based on a Bayesian statistical framework which provides a posterior probability function for each estimation. Therefore, besides the usual prediction (given in this case by the mean function), GPs come equipped with the possibility to obtain a predictive variance (i.e., error bars, confidence intervals) for each prediction. Unfortunately, the GP formulation usually assumes that there is no noise in the inputs, only in the observations. However, this is often not the case in earth observation problems where an accurate assessment of the measuring instrument error is typically available, and where there is huge interest in characterizing the error propagation through the processing pipeline. In this letter, we demonstrate how one can account for input noise estimates using a GP model formulation which propagates the error terms using the derivative of the predictive mean function. We analyze the resulting predictive variance term and show how they more accurately represent the model error in a temperature prediction problem from infrared sounding data.**

***Index Terms*— Derivative, error, Gaussian process (GP) regression, noisy, surface temperature, variance.**

I. Introduction

THE land and sea surface temperature of the earth is one of the most important components to understanding the governing physical processes on the earth's system [1]. Derived processes such as heat-fluxes and energy balances on a large temporal and spatial scale are useful for applications within climate change, vegetation monitoring, and other environmental studies. In order to acquire a complete model for many of these applications, one needs a good characterization of temperature on a global scale [2]. Acquiring ground measurements is not always practical at such a high spatial and temporal resolution scale. Remote sensing has proven to be useful to collect input data for models that capture temperature and other important environmental factors. Instruments such as the infrared atmospheric sounding interferometer (IASI) [3] have an objective to support numerical weather prediction (NWP) models to provide high-quality predictions for temperature, humidity, and some trace gases and generating global maps from satellite acquisitions which require fast and accurate algorithms.

A convenient complement to physical or numerical model inversion consists of using machine learning (ML) algorithms to substitute or emulate one or more parts in the processing pipeline. Gaussian processes (GPs) are examples of nonparametric regression models that have grown in popularity over the past decade [4]. Their strength comes from the use of Bayesian statistics in order to produce mean predictions with confidence intervals. In recent years, GPs have been successful for modeling inputs and outputs on a wide variety of tasks in remote sensing and geosciences [5].

However, one crucial limitation of many ML algorithms in general (including GPs) is their ability to handle noisy inputs. Although this is rarely studied in the ML community, this relationship is very important in the earth science and remote sensing in particular. It is customary to estimate a function $f(\mathbf{x})$ given some noisy observations at $\mathbf{y}$ for some input sample/location $\mathbf{x}$. Many ML models assume that the inputs $\mathbf{x}$ are noise-free and tailor their training procedure around this assumption. For some applications this is a valid assumption; however, as the number of data points increases and originate from different sources including other models, both assumptions become invalid and this can lead to poor modeling performance and misleading conclusions in error and uncertainty propagation studies. Quantifying uncertainty in ML models is becoming more and more prevalent despite this already being an essential part of the physical model pipeline process. On top of this, as more and more ML algorithms are being used to replace physical models, the proper analysis of the error propagation through the whole model becomes imperative.

In this letter, we apply a GP regression formulation which allows one to account for input noise estimates by exploiting the derivative of the predictive mean function. By using this method which propagates the input error, we demonstrate that the uncertainty estimates are more credible and more accurately estimate the residual error. Section II reviews the current literature on uncertain inputs in relation to GPs and introduces the proposed predictive variance estimate. Section III gives the experimental results using IASI data to predict surface temperature. Finally, Section V concludes this letter and offers some possible extensions.

II. Gaussian Processes With Noisy Inputs

This section introduces the problem of dealing with noisy inputs, reviews the theory of GP regression, and introduces the model we use to account for the error in the inputs.

Manuscript received January 14, 2019; revised April 23, 2019; accepted June 1, 2019. This work was supported by the European Research Council (ERC) through the ERC-CoG-2014 SEDAL Project under Grant 647423. *(Corresponding author: Juan Emmanuel Johnson.)*

The authors are with the Image Processing Laboratory (IPL), Universitat de València, 46010 Valencia, Spain (e-mail: juan.johnson@uv.es; valero.laparra@uv.es; gustau.camps@uv.es).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2019.2921476

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

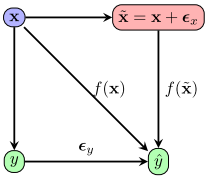


Fig. 1. Conceptual map illustrating the relation between the noisy inputs and noisy outputs for an arbitrary model. Traditional ML methods use the path from $\mathbf{x}$ to $\hat{y}$. In this letter, we investigate the use of the path from $\tilde{\mathbf{x}}$ to $\hat{y}$.

A. Regression With Uncertain Inputs

In all facets of modeling, we are essentially looking for relations between some input $\mathbf{x}$ and some output y. Using the standard ML formulation, we construct our model $f(\mathbf{x})$ plus some noise. In GPs, this noise is actually modeled as a normal distribution with some variance σ_y^2. We are interested in using a GP model under the assumption that $\mathbf{x}$ has some noise in the inputs ϵ_x as shown in Fig. 1.

B. Literature Review

In the literature, an early look at dealing with uncertain inputs is known as errors-in-variables regression [6]. In the more recent literature specifically related to GPs, we can divide the field into two families. The first family of methods adds complexity to the output noise model. This transforms the noise variance from a single scalar parameter, σ_y^2 to a more complex parameter that varies with respect to the output residuals. Using heteroscedastic noise models [7], they model the noise directly in the training and testing phase. However, it is an approximation and does not explicitly exploit the structure of the input noise and yet adds more degrees of freedom to the learning algorithm.

The second family of methods try to improve the GP model by directly considering the input uncertainty. Many of the following methods described below have applications in dynamical problems involving time series. They used local approximations of the test points from the posterior of GP function by using the Taylor approximation of the predictive mean and variance functions. Observations from [8] note that derivatives of GPs are also GPs, so the derivatives have a closed-form mean and variance function. However, their approach resulted in a minimum of second-order derivatives of the covariance function and third order to calculate the gradients for minimizing the maximum likelihood. This adds a significant layer of model and computational complexity which is problematic with large-scale problems. Another problem with this approach is that integrating over all possible trial points may not result in a Gaussian distribution. One can assume the resulting distribution of the GP model to be Gaussian and compute its mean and variance by using the approximate moments approach. This was started by [9], [10] and further developed in [11] and [12]. These approximations assume some noise in the $\mathbf{x}$ inputs but only take them into account during the predictions. Dallaire *et al.* [13] apply an uncertainty incorporating covariance function used in the training phase and similarly in [14]. However, this case is only applicable if the error in the inputs is explicitly known and does not make adjustments to the model from the posterior information. The noise-input GP (NIGP) method [15] constructs a GP framework that takes into account the posterior data by using a combination of the gradient of the predictive variance function and the input error covariance matrix. This method processes the input noise through the Taylor expansion and adds a corrective term which includes the derivative.

In remote sensing applications, typically we have well-characterized measurement errors by way of sensor error estimates during the design phase. This eliminates the need to model the input error which alleviates the computational cost of finding the value of these parameters. Although all of the above methods incorporate input noise into the GP model, the ideal situation is to account for input noise in the training procedure as done in the NIGP. However, training these models is computationally infeasible when dealing with a large amount of multidimensional data, such as in remote sensing. In the following section, we introduce and analyze a simple way to better estimate the error propagation of the test points. The approach is inspired on the NIGP formulation but adapted from the GPs used in the dynamical systems framework to suit remote sensing applications.

C. Classical GP

Let us fix the notation and the classical GP formulation first. We are given N pairs of input–output points, $\{\mathbf{x}_i, y_i | i = 1, \ldots, N\}$, where $\mathbf{x} = [x^1, \ldots, x^D]^\intercal \in \mathbb{R}^{D\times 1}$ and $y \in \mathbb{R}$. Let us define $\mathbf{X} = [\mathbf{x}_1 | \cdots | \mathbf{x}_n]^\intercal \in \mathbb{R}^{N\times D}$ be a set of known N data points, and $\mathbf{y} = [y_1, y_2, \ldots, y_n]^\intercal \in \mathbb{R}^{N\times 1}$ be the known N labels in $\mathbb{R}^{N\times 1}$. We are interested in finding a latent function $f(\mathbf{x})$ of input $\mathbf{x}$ that approximates y. We assume the function $f(\cdot)$ is corrupted by some noise

$$y = f(\mathbf{x}) + \epsilon_y \tag{1}$$

where ϵ_y represents the modeling error or residuals. By assuming a Gaussian prior for the noise term $\epsilon_y \sim \mathcal{N}(0, \sigma_y^2)$, and a zero mean GP prior for the latent function, $f(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K}$ is the covariance matrix parameterized using a Kernel function, $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, we can analytically compute the posterior distribution over the unknown output y_*, with the following predictive mean and variance for a new incoming test input point $\mathbf{x}_*$

$$\mu_{\text{GP}} = \mathbf{k}_*^\intercal (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{y} = \mathbf{k}_*^\intercal \alpha \tag{2}$$

$$v_{\text{GP}}^2 = k_{**} - \mathbf{k}_*^\intercal \left(\mathbf{K} + \sigma_y^2 \mathbf{I}_N\right)^{-1} \mathbf{k}_* \tag{3}$$

where $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), k(\mathbf{x}_*, \mathbf{x}_2), \ldots, k(\mathbf{x}_*, \mathbf{x}_N)]^\intercal \in \mathbb{R}^{N\times 1}$ calculates the similarities between the test point $\mathbf{x}_*$ and all of the training samples, $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$ is the self-similarity matrix for the test samples, and $\mathbf{I}_N$ is an identity matrix of size $N \times N$.

D. GP Regression With Noisy Inputs

To account for noisy inputs, we can restart the GP formulation under the assumption that $\tilde{\mathbf{x}}$ is the real vector that contains the real observed input $\mathbf{x}$ corrupted by some ϵ_x as in Fig. 1. By introducing $\tilde{\mathbf{x}}$ in (1), we can obtain the following model

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

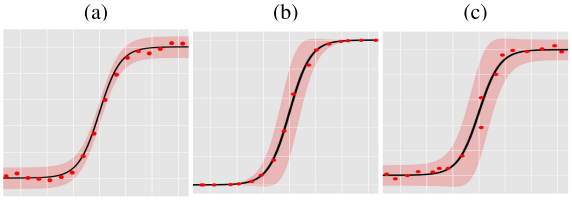


Fig. 2. Effects of considering (a) only noise for the outputs y, (b) only noise for the inputs x, and (c) noise for both the inputs x and the outputs y.

which includes the input noise in the latent function $f(\tilde{\mathbf{x}})$ which is corrupted by two sources of noise ϵ_y and ϵ_x

$$y = f(\mathbf{x} + \epsilon_x) + \epsilon_y$$

The (first-order) Taylor expansion centered at $\mathbf{x}$ provides us with a similar formulation

$$y \approx f(\mathbf{x}) + \epsilon_x^\intercal \partial_{\bar{f}} + \epsilon_y.$$

This result gives us insight into the *error propagation* of the input noise by using the derivatives of the mean function. One may imagine that in regions where there is a high gradient (where the output value is rapidly changing), the input measurements corrupted by noise will be more important as opposed to regions where the output is almost constant with regard the input. Fig. 2 illustrates this effect.

The same Gaussian prior for the noise term ϵ_y and the zero mean GP prior for the latent function $f(\mathbf{x})$ is used, like in the standard GP formulation. We can assume an additive white Gaussian noise for the inputs, $\epsilon_x \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x)$. Now we have a typical problem since we cannot analytically compute the posterior distribution of the unknown output because now our $\mathbf{x}$ comes from a distribution itself, $\mathbf{x} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x)$ [16] resulting in a non-Gaussian distribution. However, we can simply take the expectation and variance of our new function [15], [17] to be approximated as a Gaussian. The expectation gives us the same sample GP prior mean, but the resulting equation for the variance of the unknown outputs y_* for a new incoming test input point $\mathbf{x}_*$ changes as

$$v_{eGP}^2 = T_{**} + k_{**} - \mathbf{k}_*^\intercal \left(\mathbf{K} + \sigma_y^2 \mathbf{I}_N + \mathbf{T}\right)^{-1} \mathbf{k}_* \qquad (4)$$

where the effect of the noise in the inputs is represented by $\mathbf{T}_{ij} = T(\mathbf{x}_i, \mathbf{x}_j) = \partial_i^\intercal \boldsymbol{\Sigma}_x \partial_j$, and $T_{**} = T(\mathbf{x}_*, \mathbf{x}_*)$. We denoted the vector of partial derivatives of f with respect to the sample x_i as:

$$\partial_i := \left[\frac{\partial f(\mathbf{x}_i)}{\partial x_i^1} \cdots \frac{\partial f(\mathbf{x}_i)}{\partial x_i^D}\right]^\intercal.$$

The derivative of the predictive function f (2) in GPs only depends on the derivative of the Kernel function since it is linear with respect to the $\boldsymbol{\alpha}$ parameters

$$\frac{\partial f(\mathbf{x}_i)}{\partial x_i^j} = \frac{\partial \mathbf{k}_i \boldsymbol{\alpha}}{\partial x_i^j} = (\partial \mathbf{k}_{ij})^\intercal \boldsymbol{\alpha}$$

where $\partial \mathbf{k}_{ij} = [(\partial k(\mathbf{x}_i, \mathbf{x}_1)/\partial x_i^j), \ldots, (\partial k(\mathbf{x}_i, \mathbf{x}_N)/\partial x_i^j)]^\intercal$. Please see[1] for a working implementation of the error GP (eGP) model.

[1] https://isp.uv.es/egp.html

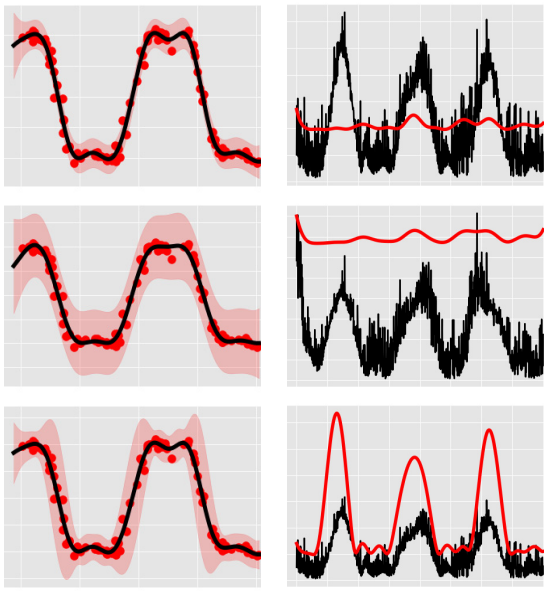


Fig. 3. (Top): illustrative unidimensional example showing the predictive mean and predictive variance using different GP methods: standard GP. (Middle): GP with heteroscedastic noise levels. (Bottom): GP with variance moment matching error propagation. The left column shows the predictive mean and the predictive variance for each method. The right column shows the empirical variance (red line) in relation to the empirical predictive variance (black line). An input noise coefficient of $\sigma_x = 0.3$ and an output noise coefficient $\sigma_y^2 = 0.05$ was used for this demonstration.

III. Experiments

A. Illustrative Example

Fig. 3 showcases a simple "nearly-square" sine wave and how three different GP formulations approximate the variance of the function. A standard GP (top row) clearly does not have any correspondence to the errors in the inputs as the line is nearly constant for all regions. A GP using a heteroscedastic noise model (middle row) does capture and adjust for the input noise but the output still remains constant with no confidence in regions where the outputs do not vary. On the bottom row, the GP with our adjusted variance correctly adjusts the predictive variance based on regions where the gradient is higher and the outputs drastically change. This same response to the input noise is what we are hoping to accomplish with a real and more complex data set in Section III-B.

B. Temperature Estimation From Infrared Sounding Data

We illustrate how the proposed predicted variance accounting for the input errors (eGP) compares to the standard predictive variance for typical GP models when trained to estimate surface temperature from noisy input radiance values.

1) Data: We use data acquired by the IASI instrument onboard the MetOp-A satellite, which consists of 8461 spectral channels between 3.62 and 15.5 μm with a spectral sampling of 0.25 cm^{-1} and a spatial resolution of 25 km. We chose the October 1, 2013, for our sample space, which contained 13 complete orbits within a 24-h period. Since temperature is

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

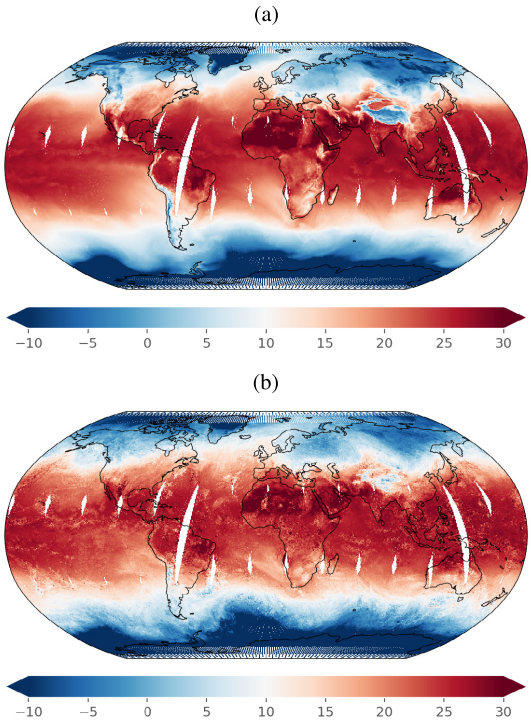


Fig. 4. Mean temperature prediction for October 1, 2013. (a) Ground truth from the ECMWF model. (b) Predictions using the predictive mean of the GP.

an exemplary atmospheric parameter for weather forecasting, we used the atmospheric surface temperature predictions of the European Centre for Medium-Range Weather Forecasts (ECMWF) model and the radiance IASI measurements. The IASI instrument is well characterized and the noise can be described as an additive Gaussian noise with a covariance error for the radiance values provided by European Organization for the Exploitation of Meteorological Satellites (EUMETSAT).

2) Methodology: We have 13 orbits in total. $N_{\text{train}} = 5000$ points were selected as training samples randomly from all orbits. The extremely noisy channels were discarded to reduce the dimensionality of the data from 8461 to 4699. This was followed by principal component analysis (PCA) to further reduce the number of dimensions from over 4699 to 50 accounting for 99% of the variance within the data. Using this reduced sample space, we train a standard GP model using a negative maximum-log-likelihood scheme with ten random optimizer restarts for a standard radial basis function (RBF) Kernel. The remaining points, total of $N_{\text{test}} = 1\,182\,600$ points, were used for testing. We use the same standard GP to calculate the predictive mean of for the test points. We calculate the standard deviation (3) and our augmented standard deviation with input variances (4) to compare to the mean absolute error.

3) Temperature Estimation: Statistically, the estimation using the predictive mean of the trained GP model achieved an average mean absolute error $e_{\text{GP}} = |y - \mu_{\text{GP}}|$ of around

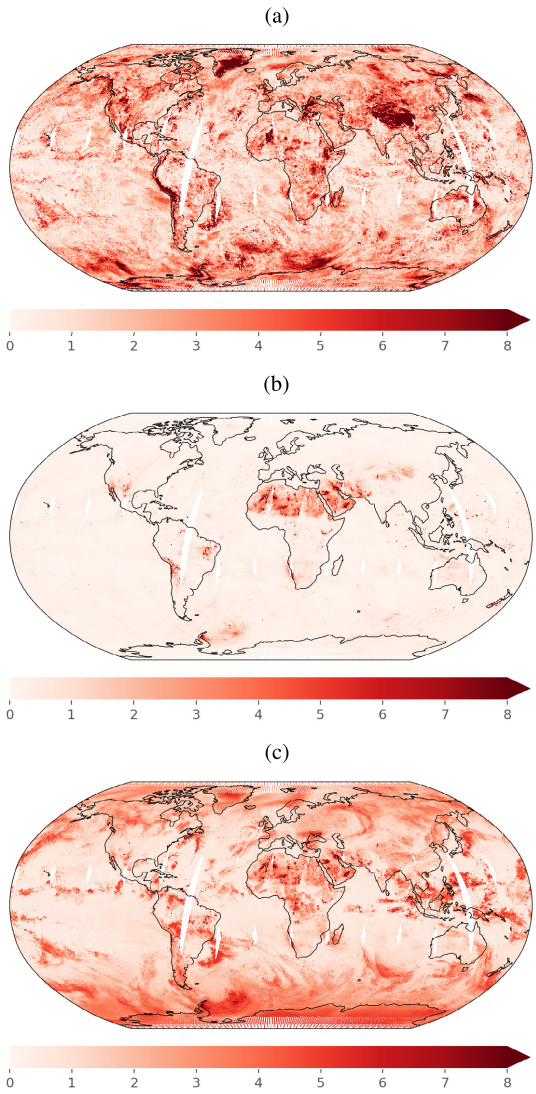


Fig. 5. Errors and standard deviations of our GP model for the mean temperature with orbit October 1, 2013. (a) Absolute error, e_{GP}, between our predictions and the ECMWF model. (b) Standard deviation, v^2_{eGP}, from a regular GP (3). (c) Standard deviation, v^2_{eGP}, which takes into account the noise in the inputs, eGP (4).

2 °C and a model R^2 value of 0.97. The first set of temperature maps in Fig. 4 show the mean surface temperature ground truth provided by ECMWF model versus the GP model predictions. Visibly, the results are similar but there are some discrepancies in regions where there is a large change in temperature. For example regions along the boundaries between the red and blue in the southern and northern regions of the equation exhibit errors in predictions. Furthermore, the boundary along the west coast of south America, regions near the equator and tropic of Capricorn, and the region in central Asia have different temperature predictions than the ground truth. Fig. 5(a)

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I

METRICS BETWEEN THE **MEAN ABSOLUTE ERROR** (e_{GP}) OF THE GP MODEL AND THE **PREDICTIVE STANDARD DEVIATION** FOR BOTH THE GP ν^2_{GP} (3) AND eGP ν^2_{eGP} (4)

Statistic	ν^2_{GP}		ν^2_{eGP}	
	Mean	Variance	Mean	Variance
Mean Absolute Error	1.92	3.05	**1.20**	**1.63**
Mean Squared Error	7.90	81.91	**4.10**	**80.85**
Root Mean Squared Error	2.81	9.05	**2.05**	**9.00**

supports these visible observations as the reddest regions are the same, and also has red-regions along the northern and southern hemispheres.

4) Error Propagation: The maps for the two GP variance predictors are noticeably different [Fig. 5(b) and (c)]. The standard GP variance focuses on the region in northern Africa and the middle east. This is an absent region in the error map. This is also a known hot region which is expected to have a high temperature gradient locally but relatively similar temperature gradient spatially. The standard deviation for the eGP also has a large confidence interval for this area, but, in addition, chooses regions where there is a high temperature gradient spatially, like the aforementioned regions in the northern and southern hemispheres. Many of the slightly less red regions correspond to the regions on the error map. Overall, it appears that the spread between the regions with lower and higher standard deviations is more pronounced in the eGP than the standard GP which has a high concentration on regions where the temperature is spatially similar.

Table I shows the numerical results. It presents the error estimates between the standard deviation for the GP and eGP versus the mean absolute error between the predictions and the labels. In all cases, the eGP and the absolute error have lower error statistics.

IV. CONCLUSION

The consideration of noisy inputs is extremely important in earth science for error characterization and uncertainty quantification and propagation. However, their formal treatment in ML has not been widely approached. If we hope to combine the use of statistical models with physical models, then we will need accurate error and uncertainty estimates for our predictions.

In this letter, we gave a simple formulation and rationale for how the derivative of GP models in particular can be used to help the predictive variance obtain more accurate (and credible) error estimates in earth science applications. Using a GP model to predict temperature from radiances, we showed quantitatively and visually that the predictive variance with the propagated error provided a stronger correspondence to the absolute error and that it can be useful to understand a GP models performance and noise/errors impact.

For further work, one could incorporate the input noise information both during the training procedure of the GP algorithm and in the computation of the predictive variance in the testing procedure, as well as incorporate this same framework of utilizing the derivatives of Kernel functions to propagate the error in other Kernel methods.

REFERENCES

[1] D. Blondeau-Patissier, J. F. R. Gower, A. G. Dekker, S. R. Phinn, and V. E. Brando, "A review of ocean color remote sensing methods and statistical techniques for the detection, mapping and analysis of phytoplankton blooms in coastal and open oceans," *Prog. Oceanogr.*, vol. 123, pp. 123–144, Apr. 2014.

[2] Z.-L. Li *et al.*, "Satellite-derived land surface temperature: Current status and perspectives," *Remote Sens. Environ.*, vol. 131, pp. 14–37, Apr. 2013.

[3] T. August *et al.*, "IASI on Metop-A: Operational level 2 retrievals after five years in orbit," *J. Quant. Spectrosc. Radiat. Transf.*, vol. 113, pp. 1340–1371, Jul. 2012.

[4] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes in Machine Learning* (Adaptive Computation and Machine Learning). Cambridge, MA, USA: MIT Press, 2005.

[5] G. Camps-Valls, J. Verrelst, J. Munoz-Mari, V. Laparra, F. Mateo-Jimenez, and J. Gomez-Dans, "A survey on Gaussian processes for earth-observation data analysis: A comprehensive investigation," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 58–78, Jun. 2016.

[6] P. Dellaportas and D. A. Stephens, "Bayesian analysis of errors-in-variables regression models," *Biometrics*, vol. 51, no. 3, pp. 1085–1095, 1995.

[7] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian process regression," in *Proc. 24th Int. Conf. Mach. Learn.*, New York, NY, USA, 2007, pp. 393–400.

[8] E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen, "Derivative observations in Gaussian process models of dynamic systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1057–1064.

[9] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 15, Dec. 2003, pp. 545–552.

[10] A. G. Quinonero-Candela and C. Rasmussen, "Prediction at an uncertain input for Gaussian processes and relevance vector machines application to multiple-step ahead time-series forecasting," Institutive Mathemetical Model., DTU, Chennai, India, Tech. Rep. 1, 2003.

[11] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic moment-based Gaussian process filtering," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 225–232.

[12] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 465–472.

[13] P. Dallaire, C. Besse, and B. Chaib-Draa, "An approximate inference with Gaussian process to latent functions from uncertain data," *Neurocomputing*, vol. 74, no. 11, pp. 1945–1955, 2011.

[14] A. Girard and R. Murray-Smith, "Learning a Gaussian process model with uncertain inputs," Dept. Comput. Sci., Univ. Glasgow, Glasgow, U.K., Tech. Rep. TR-2003-144, 2003.

[15] A. Mchutchon and C. E. Rasmussen, "Gaussian process training with input noise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1341–1349.

[16] J. Quiñonero-Candela and S. T. Roweis, "Data imputation and robust training with Gaussian processes," Tech. Rep., 2003.

[17] McHutchon, "Nonlinear modelling and control using Gaussian processes," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2014.

D.3. Paper III

GAUSSIANIZING THE EARTH – MULTIDIMENSIONAL INFORMATION MEASURES FOR EARTH DATA ANALYSIS

A PREPRINT

J. Emmanuel Johnson[*]
Image Processing Laboratory
Universitat de València
València, Spain
juan.johnson@uv.es

Valero Laparra[†]
Image Processing Laboratory
Universitat de València
València, Spain
valero.laparra@uv.es

Maria Piles[‡]
Image Processing Laboratory
Universitat de València
València, Spain
maria.piles@uv.es

Gustau Camps-Valls[§]
Image Processing Laboratory
Universitat de València
València, Spain
gcamps@uv.es

November 26, 2020

ABSTRACT

Information theory is an excellent framework for analyzing Earth system data because it allows us to characterize uncertainty and redundancy, and is universally interpretable. However, accurately estimating information content is challenging because spatio-temporal data is high-dimensional, heterogeneous and has non-linear characteristics. In this paper, we apply multivariate Gaussianization for probability density estimation which is robust to dimensionality, comes with statistical guarantees, and is easy to apply. In addition, this methodology allows us to estimate information-theoretic measures to characterize multivariate densities: information, entropy, total correlation, and mutual information. We demonstrate how information theory measures can be applied in various Earth system data analysis problems. First we show how the method can be used to jointly Gaussianize radar backscattering intensities, synthesize hyperspectral data, and quantify of information content in aerial optical images. We also quantify the information content of several variables describing the soil-vegetation status in agro-ecosystems, and investigate the temporal scales that maximize their shared information under extreme events such as droughts. Finally, we measure the relative information content of space and time dimensions in remote sensing products and model simulations involving long records of key variables such as precipitation, sensible heat and evaporation. Results confirm the validity of the method, for which we anticipate a wide use and adoption. Code and demos of the implemented algorithms and information-theory measures are provided.

***K*eywords Density estimation, Information theory, entropy, mutual information, Gaussianization, multivariate data, droughts, climate extreme, anomaly, spatio-temporal Earth data.**

[*] https://jejjohnson.netlify.app
[†] https://www.uv.es/lapeva/
[‡] https://sites.google.com/site/mariapiles/
[§] https://www.uv.es/gcamps/

1 Introduction

Earth system models and observational data are fundamental to monitor our planet and to understand climate change [1, 2, 3, 4]. We now face a data deluge which comes from remote sensing platforms that continuously increase the spatial, temporal and spectral resolution of data sources. In recent years, Earth system data comes in high volume, heterogeneity, and uncertainty [5] which poses important challenges in analysis, modeling and understanding. The statistical analysis of remote sensing data and model simulations requires dealing with large amounts of heterogeneous, multivariate, and spatio-temporal data. The volume of data from high-resolution models and observations have substantially increased to petabyte scales. Yet, we are well aware that copious amounts of data does not necessarily mean large amounts of *information*. For example, it is now widely acknowledged that models are often correlated and share common traits, features and information content. Which is the most appropriate and representative model? How can we best quantify their information content in meaningful units? Essential Earth variables and data products exhibit high levels of redundancy in space and time. What is the appropriate space, time or spatio-temporal scales one should look at? The same questions arise when trying to assess and choose the most adequate observational variable or bio-geo-physical parameter for Earth monitoring.

Density estimation is at the core of all problems in statistics and machine learning, but is still a challenging and unresolved problem.

From a pure statistical standpoint, the problem of *information quantification* for Earth and climate data is challenging. Information theory (IT) is the appropriate framework to study information content, uncertainty and redundancy [6]. The estimation of entropy and mutual information for discrete and continuous random variables has been addressed under different approaches in the statistics literature [7, 8, 9, 10]. An important problem is that IT measure estimation of multivariate data is difficult, and very often only unidimensional/marginal measures of information are computed in practice. Many are largely based on histogram estimates, which is a very limiting factor [6, 11]. Many multivariate estimates based on nearest neighbors [8, 9, 10] either do not scale well, do not converge to the true measure, or show high estimation bias [12]. Measures such as entropy or mutual information have been used in remote sensing and the geosciences to study feature redundancy in image classifiers [13], to assess the maximum number of parameters that can be estimated given a set of observations [14], for remote sensing feature extraction and weighting [15, 16], data fusion [17], image registration [18, 19, 20], and to quantify uncertainty in models and observations [21].

Information quantification requires estimating multivariate densities which is a challenging and unresolved problem. This is especially problematic in Earth observation data with moderate-to-high dimensional problems with nonlinear feature relations. These issues affect the classic parametric density estimators based on the exponential family of solutions or mixture distributions as well as non-parametric methods based on histograms, kernel density estimation, and k-nearest neighbors. As an alternative to these traditional methods, there is a new class of methods called neural density estimators [22] which are parameterized neural networks that estimate densities. They use the 'change of variables' formula to estimate densities of inputs and also allow one to draw samples of your input data. They have promise as they have been successfully used in applications related to Earth system sciences including inverse problems [23] and density estimation [24].

Gaussianization is effective in multivariate density estimation, and allows one to estimate multi-dimensional information measures

In this paper, we look at a particular class of models inside the neural density estimation family. In particular, we introduce the Gaussianization method [25] and in particular a generalized algorithm called Rotation-Based Iterative Gaussianization (RBIG) [26]. This method uses a repeated sequence of simpler feature-wise Gaussian transformations and orthogonal rotations until convergence. It can be shown that in each iteration the total correlation and the non-Gaussianity are reduced and converge towards zero, that is, towards full independence. The learned transformation towards the Gaussian domain is invertible, which allows us to synthesize data easily by inverting samples drawn from the Gaussian domain. The method is also advantageous because it allows us to estimate IT measures such as entropy, total correlation, non-Gaussianity and mutual information effectively in high dimensional data. The method is easy to apply, fast, and has links to deep neural networks [26, 27, 28]. Section §2 will review the theoretical properties of RBIG and their practical use for information theoretic measures estimation.

In §3 we take advantage of RBIG to estimate information theoretic measures in different Earth system science problems of interest. Three settings of increasing scale and sophistication are given (cf. Table 3): from working at pixel and patch level (fully spectral and spatio-spectral domains) to studying information in time series (fully temporal domain), and finally to quantifying redundancy and probability in Earth data cubes (spatio-temporal domain). We first illustrate the use of RBIG with three standard remote sensing data modalities and for three different illustrative applications: in Gaussianizing radar backscattering intensity data, synthesizing hyperspectral spectra and quantifying information in RGB aerial images. Our second application is concerned with assessing the information content conveyed by a

selection of remotely-sensed variables widely used in vegetation/land monitoring -temperature, moisture and vegetation indices-, and with investigating the temporal scales that maximize their shared information under extreme events such as droughts. Finally, we focus on quantifying the information content in spatio-temporal data cubes of selected climate variables (precipitation, sensible heat, evaporation) over a decade of global data. We are interested in quantifying and contrasting the information content of the space versus time dimensions as a means to understand the scales of the underlying physical processes. We conclude with some remarks and outline further work in §4.

2 Multivariate Gaussianization

2.1 PDF Estimation

Most problems in signal and image processing, information theory and machine learning involve the challenging task of multidimensional probability density function (PDF) estimation. A probability density function or simply a density $p(\cdot)$ takes an input $\mathbf{x} \in \mathcal{X}$ and outputs a density, which follows the properties that 1) $p(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathbb{R}^D$, and 2) it has to sum to one, $\int_{\mathcal{X}} p(\mathbf{x})\mathrm{d}\mathbf{x} = 1$. In practice, we usually do not have access to the PDF $p(\cdot)$ but we do have a set of (multivariate) samples drawn from the generating process $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ to estimate the PDF from. An accurate PDF estimation is important because it allows us to: 1) calculate the probability of any arbitrary input data point, which accounts for the relative likelihood that the value of the random variable (r.v.) would equal the sample; 2) generate samples $\mathbf{x}' \sim p(\mathbf{x})$ from this distribution thus allowing data synthesis, background and support estimation, as well as anomaly detection; and 3) calculate expectations for functions (or transformations) of arbitrary form $f(\mathbf{x})$ given $p(\mathbf{x})$, i.e. $\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})]$, which allows us to e.g. characterize the system.

Having access to all of these properties gives us the ability to tackle long-standing problems in machine learning and statistics. With accurate PDF estimates, one can model conditional densities of data generated from a prior distribution, develop accurate and efficient compression schemes, and use principled objective functions such as maximum likelihood. In addition, having access to an accurate density estimator can be useful in many hybrid applications to deal with out-of-sample or out-of-distribution problems too [29]. The problem is therefore to estimate the density $p(\mathbf{x})$ given a set of samples from $\mathcal{X}$.

The simplest approach to PDF estimation assumes the density has a *parametric* functional form defined by a fixed number of tuneable parameters. The Gaussian assumption is the most widely adopted for unimodal distributions, which comes parameterized by a mean $\boldsymbol{\mu}$ and a covariance function $\boldsymbol{\Sigma}$. If more than one mode is assumed, then a mixture of Gaussians generally leads to better fits. However, finding a parametric form for the distribution that fits properly to a particular data is really difficult in most cases.

The alternative approach comes from *non-parametric models*, which do not assume a specific form for the distribution and are learned from data. The simplest non-parametric method estimates the PDF by partitioning the data space in non-overlapping bins whereby the density is estimated as the fraction of data points in the bin divided by the volume of the bin. This histogram-based PDF estimation method poorly copes with dimensionality, typically leads to either overfitting or underfitting, and selecting an appropriate number of bins per dimension is a challenge in itself. Alternative parametric estimates for these methods following likelihood-estimation schemes for the optimal bin width determined by the maximum likelihood have been introduced [22]. However, they are very *rigid* approaches and lead to very rough density functions. To achieve *smoother* PDF estimates, the kernel density estimation (KDE) method is another popular non-parametric method. KDE places a non-linear kernel function with a varying bandwidth parameter to control the degree of smoothness on top of each example. Unfortunately, a bias-variance trade-off will result in over/underfitting the PDF, especially in moderate-to-high dimensional problems. In the previous approaches, the bandwidth is typically fixed *a priori* following heuristics in the literature [30], and rarely take into account the concentration of points, i.e. that smaller bins should be placed in regions with a higher concentration of points, in a form of adaptive bit-allocation scheme. This can be addressed by using k-nearest neighbors (kNN), which has an *adaptive* bandwidth per location and depends on the number of training points available. However, all of the density estimators above suffer from the curse of dimensionality: as the dimensionality increases, the space becomes sparser and density estimates are unreliable.

2.2 Gaussianization for PDF estimation

An alternative way to estimate a PDF from observational data is through a data transformation to a *convenient* domain, instead of working explicitly in the high-dimensional input domain. The question of what is a convenient domain is a long-standing one, yet ideally this should be a domain with independent components so one can work in each dimension independently to get rid of the curse of dimensionality, one that allows to perform operations and compute quantities therein, and one that is invertible so that one can express these quantities in meaningful units of the input domain.

Table 1: Summary of all components of the Gaussianization algorithm.

Description	Notation	Transformation	Domain	Before	After
Marginal Uniformization	U	Histogram [26], Kernel Density Estimation [28], Lambert[31], Splines[32], Box-Cox [33]	$\mathbb{R} \rightarrow \mathbb{R}^{[0,1]}$		
Inverse CDF	CDF^{-1}	Inverse Gaussian CDF, Logit, Inverse Cauchy CDF	$\mathbb{R}^{[0,1]} \rightarrow \mathbb{R}$		
Marginal Gaussianization	$\Psi = \text{CDF}^{-1} \circ \text{U}$	Marginal Uniformization + Inverse CDF	$\mathbb{R} \rightarrow \mathbb{R}$		
Rotation	$\mathbf{R}$	Principal Components Analysis [26], Independent Components Analysis [25], Random Rotations [26]	$\mathbb{R}^d \rightarrow \mathbb{R}^d$		
Gaussianization Block	$G_\ell = \mathbf{R}\left[\Psi^1 \cdots \Psi^d\right]$	Composition of Rotation + Marginal Gaussianization	$\mathbb{R}^d \rightarrow \mathbb{R}^d$		
Gaussianization transform	$G = [G_1 \circ \cdots \circ G_L]$	Composition of Gaussianization Blocks	$\mathbb{R}^d \rightarrow \mathbb{R}^d$		

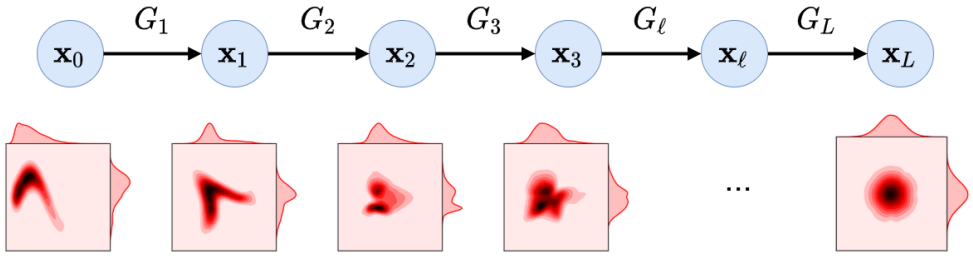


Figure 1: A demonstration of a complete Gaussianization of a noisy sine wave to a marginally and jointly Gaussian distributed. We use PCA for the rotation matrix and histogram CDF estimator for the marginal transformation.

The Gaussian distribution has desirable properties of showing independent components and being mathematically tractable and is thus a good candidate for density estimation. A class of Gaussianization methods [28, 26] look for transforms to a multivariate Gaussian domain. These transforms are related to projection pursuit transformations originally introduced in [34] and seek to transform a multivariate distribution $p(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$, into a standardized multivariate Gaussian distribution [25, 26]:

$$G_{\boldsymbol{\theta}}: \begin{array}{ccc} \mathbf{x} \in \mathbb{R}^d & \mapsto & \mathbf{z} \in \mathbb{R}^d \\ \sim p(\mathbf{x}) & & \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \end{array} \tag{1}$$

where $\boldsymbol{\theta}$ are the parameters learned to Gaussianize the data $\mathbf{x}$, $\mathbf{0}$ is a vector of zeros (for the means) and $\mathbf{I_d}$ is the identity matrix (for the covariance). By construction, the Gaussianization transform is a parameterized function $G_{\boldsymbol{\theta}}$ consisting of a sequence of L iterations (or layers), each one of them performing an orthogonal rotation of the data and a marginal Gaussianization transformation to each feature.

The transformation $G_{\boldsymbol{\theta}}$ in each iteration ℓ is defined as:

$$G_{\boldsymbol{\theta}}: \mathbf{x}_{\ell+1} = \mathbf{R}_\ell \boldsymbol{\Psi}_\ell(\mathbf{x}_\ell), \quad \ell = 1, \ldots, L$$

where $\mathbf{x}_0$ corresponds to the original data $\mathbf{x}$, $\mathbf{\Psi}_\ell$ is the marginal Gaussianization of each dimension of $\mathbf{x}^\ell$ for the iteration ℓ, and $\mathbf{R}_\ell$ is a rotation matrix for the marginally Gaussianized variable $\mathbf{\Psi}_\ell(\mathbf{x}^\ell)$. After convergence in L iterations, the transformation contains all the needed information to transform data coming from the original density into a multivariate Gaussian. $\boldsymbol{\theta}$ collectively group all parameters in the method: those from the rotation matrix $\mathbf{R}$ and the marginal transformation $\mathbf{\Psi}$. For example, one could use a principal components analysis (PCA) transformation for the rotation matrix $\mathbf{R}$ and a histogram transformation for the marginal Gaussianization transformation Ψ. Then, the eigenvectors obtained from PCA describing $\mathbf{R}$ and the parameterizations of $\mathbf{\Psi}$ would define $\boldsymbol{\theta}$. See Table 1 for more details on the decomposition of this formula and Fig. 1 for a full decomposition of a toy dataset.

We can use the change of variables formula to calculate the PDF of $\mathbf{x}$ as

$$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{z}}\left(G_{\boldsymbol{\theta}}(\mathbf{x})\right)\left|\nabla_{\mathbf{x}} G_{\boldsymbol{\theta}(\mathbf{x})}\right|, \tag{2}$$

where $|\nabla_{\mathbf{x}} G_{\boldsymbol{\theta}}(\mathbf{x})|$ is the determinant of the Jacobian of $G_{\boldsymbol{\theta}}$ w.r.t. $\mathbf{x}$. Generally, any unknown PDF of $\mathbf{x}$ can be estimated as long as we have the transformation $G_{\boldsymbol{\theta}}$ along with its Jacobian. Intuitively, this transformation essentially destroys the density of $\mathcal{X}$ into unstructured noise (often Gaussian) [35]. There is no limit to the amount of composite transformations $G_{\mathbf{\Theta}} = G_{\boldsymbol{\theta}_1} \circ G_{\boldsymbol{\theta}_2} \circ \cdots \circ G_{\boldsymbol{\theta}_L}$ that can be used in order to sufficiently converge to the Gaussian distribution. In addition, because $G_{\mathbf{\Theta}}$ is invertible, we can sample points in the original domain $\mathbf{x}' \in \mathcal{X}$ by generating samples in the transformed Gaussian domain and propagating this through the inverse transformation $G_{\mathbf{\Theta}}^{-1}$. Because the transform is a product of linear and marginal operations, both the Jacobian and the inverse transform can be computed easily [26, 36].

The transformation is invertible and allows for density estimation, synthesis and information quantification in high-dimensional Earth data problems

The original Gaussianization algorithm [25] worked by applying an orthogonal rotation matrix via independent components analysis (ICA) and then a mixture of Gaussians (MOGs) for the marginal Gaussian transformation. After enough repetitions L, it was shown that this converged to a multivariate Gaussian distribution [25]. In [26] we extended Gaussianization by realizing that the method will converge with any orthogonal rotation matrix $\mathbf{R}$ and we named the algorithm *Rotation-Based Iterative Gaussianization* (RBIG). This allowed for more simpler and faster algorithms such as the Principal Components Analysis (PCA) and even randomly generated orthogonal rotation matrices. In addition, much simpler univariate estimators like the histogram was used to speed up the algorithm significantly. Meng et. al. [28] coined the term *Gaussianization Flows* and extended the iterative algorithm to be fully parameterized and trainable by incorporating a mixture of logistics as the marginal Gaussianization layer and a sequence of Householder Flows [37, 38] as the rotation layer. They also proved this is a universal approximator and showed convincing results that Gaussianization is comparable to some other classes of methods specifically designed for density estimation or sampling [28]. All transformations and example variants can be found in table 1.

Irregardless of the method chose, in order to find the parameters $\boldsymbol{\theta}$ for the transformation $G_{\boldsymbol{\theta}}$, we minimize the following cost function w.r.t. $\boldsymbol{\theta}$:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathrm{D}_{\mathrm{KL}}\left[p_{\mathbf{z}}\left(G_{\boldsymbol{\theta}}(\mathbf{x})\right) || \mathcal{N}(\mathbf{0}, \mathbf{I}_D)\right], \tag{3}$$

which is the Kullback-Leibler (KL) divergence between the estimated Gaussian distribution and the true multivariate Gaussian distribution of mean $\mathbf{0}$ and covariance $\mathbf{I}$; in other words, this is a measure of how much *non-Gaussian* our distribution is after transformation. This reveals a direct relationship with information theoretic concepts and measures. Chen [25, 39] showed that (3) can be decomposed as

$$\mathcal{L}(\theta) = T(\mathbf{x}) + J_m(\mathbf{x}), \tag{4}$$

where $T(\mathbf{x})$ is the Total Correlation (T) (a.k.a. Multi-Information, Multivariate Mutual Information) between all of the marginal distributions, and $J_m(\mathbf{x})$ is the KL divergence between the marginal distributions and the standard Gaussian normal distribution. Intuitively, this cost function is trying to minimize the shared information between each of the marginal distributions and ensuring that they follow a standard normal Gaussian distribution. We want to highlight here that RBIG vastly transforms and simplifies the PDF estimation problem: from estimating the density of the high-dimensional multivariate distribution in $\mathcal{X}$ directly, to doing it indirectly through a transformation to a Gaussian domain. All this by using a series of marginal transformations, which are straightforward and fast.

An illustrative example of how RBIG works on a simple 2D toy dataset is shown in Figure 2. We transform a non-Gaussian 2D dataset into a 2D marginal and jointly Gaussian distribution along with the inverse transformation (first row). The second row showcases how we can use RBIG to synthesize points in the data domain using the inverse transformation. The bottom right figure shows evolution through iterations of the final total correlation (as a measure of redundancy) and the Non-Gaussianity (as a measure of distance to a Gaussian). Please see

RBIG site: https://ipl-uv.github.io/rbig/

for a working implementation of the RBIG algorithm in Python and MATLAB.

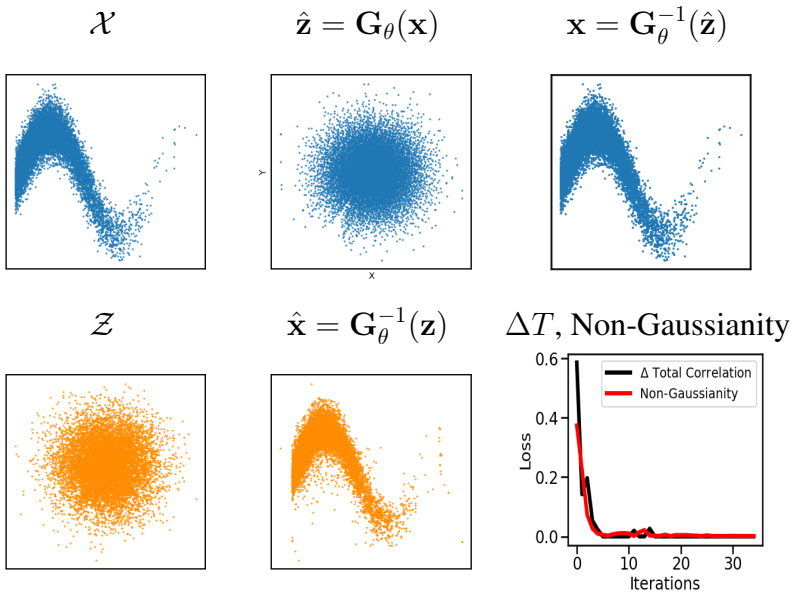


Figure 2: Density estimation of sinusoid with heteroscedastic noise using RBIG. Top: The original data distribution $\mathcal{X}$ is mapped to a Gaussian domain $\mathcal{Z}$ with transform $\mathbf{G}_\theta$ parameterized by a set of rotations and marginal Gaussianizations collectively denoted as θ, which has an analytic inverse transformation, $\mathbf{x} = \mathbf{D}_\theta^{-1}(\hat{z})$ to recover the original data. Bottom: One can sample random data from the Gaussian in domain $\mathcal{Z}$ and use the inverse transformation of $\mathbf{z}$ to $\hat{\mathbf{x}}$ for data synthesis. We also demonstrate the losses; the equivalence of the change in total correlation between layers ΔT and the KL-Divergence between transformed data and a multivariate Gaussian (Non-Gaussianity)

2.3 Information Theory Measures using the RBIG Transform

RBIG was designed for density estimation, but was inspired by and had connections to information theory [6]. The series of transformations learned by RBIG lead to a Gaussian domain so features are statistically independent. This reduction in redundancy is achieved iteratively and can be explicitly computed by summing up all the layer redundancy reductions. This metric is known as the total correlation and computing this metric subsequently allows us to derive *some information-theoretic measures* from the data.

2.3.1 Information

Shannon information I [40] is based on the idea that a sample, $\mathbf{x}_i$, is more interesting (carries more information) if it is less probable. The formal definition of information is:

$$I(\mathbf{x}_i) = -\log(p_{\mathbf{x}}(\mathbf{x}_i)). \tag{5}$$

It can be used for instance to highlight regions of more interest in a dataset. Information can be computed for each sample in our dataset by using RBIG and Eq. (2).

The expected value of the provided information by a complete dataset, $\mathbf{x}$, is called entropy:

$$H(\mathbf{x}) = \mathbb{E}_{\mathbf{x}}[-\log(p_{\mathbf{x}}(\mathbf{x}))]. \tag{6}$$

While the entropy could be computed by estimating the information of each sample in the dataset using Eq. (5) and averaging, it is more convenient to compute it by using the ability of RBIG to compute Total Correlation as we will see in the following section.

2.3.2 Total Correlation

The total correlation, T, accounts for the information shared among the dimensions of a multidimensional random variable [41, 42]. Details on how to compute T using RBIG can be found in [26], here we sketch the main idea. Given

data $\mathbf{x} \in \mathbb{R}^D$, we first learn the Gaussianization transform with L iterations, and compute the cumulative reduction in total correlation in each iteration as:

$$T(\mathbf{x}) = \sum_{\ell=1}^{L} \left(D\, H(\mathcal{N}(0,1)) - \sum_{d=1}^{D} H(\mathbf{x}_d^{\ell}) \right). \tag{7}$$

The number of layers L will be determined by the reduction in total correlation with each transformation. If there is no change in total correlation after some threshold number of layers, we can assume that x_d are completely independent. It is important to note that all entropy calculations only involved marginal operations which are simple and fast which allows RBIG to be used on large datasets with a high number of dimensions.

2.3.3 Joint entropy

While the concept of information is attached to a a particular sample, entropy is used in different fields to characterize how unpredictable a complete process is. Entropy can be easily computed from the learned RBIG transformation by

$$H(\mathbf{x}) = \sum_{d=1}^{D} H(\mathbf{x}_i) - T(\mathbf{x}), \tag{8}$$

where $\sum_{d=1}^{D} H(\mathbf{x}_i)$ are marginal entropy estimations and $T(\mathbf{x})$ also involves marginal estimations, cf. (7).

2.3.4 Multivariate Mutual Information

The multivariate mutual information (MI) accounts for the information shared by two datasets [6]. Estimating MI can be very challenging when working with high dimensional data. Our approach is based on the invariance property of mutual information to reparameterize the space of each variable [43]. Therefore, we essentially Gaussianize the two datasets, $\mathbf{X}$ and $\mathbf{Y}$, with corresponding transforms that remove their total correlations. Then, the total correlation remaining among both Gaussianized datasets is equivalent to the mutual information between the original datasets:

$$MI(\mathbf{X}, \mathbf{Y}) = T([\mathbf{G}_{\theta_{\mathbf{x}}}(\mathbf{X}), \mathbf{G}_{\theta_{\mathbf{y}}}(\mathbf{Y})]), \tag{9}$$

which again implies only marginal operations, cf. eq.(7).

Figure 3 shows a venn diagram illustrating the different information theory measures used in this paper and Table 2 shows a visual demonstration of how they compare to the popular Pearson correlation coefficient for different toy datasets.

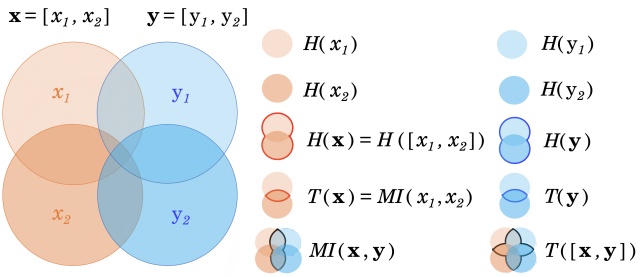
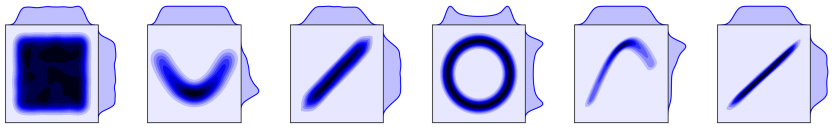


Figure 3: A venn diagram showing the relationships of all information theory measures used in this paper. The solid coloured circles represent marginal variables and the intersection regions with bold lines represent regions for information theory measures like mutual information, MI and total correlation, T

3 Experiments

In this section, we explore the information content, the redundancy and the relation in a selection of Earth data analysis problems, involving both remote sensing data and models, using RBIG. First we illustrate the method ability to analyze standard remote sensing settings involving total correlation estimation in hyperspectral, radar and very high resolution

Table 2: Demonstration showing how different information theory measures discussed compare to the popular Pearson correlation coefficient, ρ. This table is also a visual demonstration of how to interpret Mutual Information and how it's related to marginal entropy and the joint entropy; $MI(\mathbf{x}, \mathbf{y}) = H(\mathbf{x}) + H(\mathbf{y}) - H(\mathbf{x}, \mathbf{y})$



Correlation	$\rho(\mathbf{x}, \mathbf{y})$	Low	Medium	Low	Low	Medium	High
Mutual Information	$MI(\mathbf{x}, \mathbf{y})$	Low	Medium	High	High	High	High
Marginal Entropy	$H(\mathbf{x}), H(\mathbf{y})$	High	High	High	High	High	High
Joint Entropy	$H(\mathbf{x}, \mathbf{y})$	High	Medium	Medium	Low	Low	Low

Table 3: Summary of experiments with details on the data sets, configurations, application and measures employed.

Exp.	Data sets	Characteristics	Ref.	Configuration	Application	Measures
1	SAR: ERS-2	26m, backscatter intensity	[44]	Pixel-wise	Gaussianization	T
	Hyperspectral: AVIRIS	30m, 224 channels	[45]	Pixel-wise	Synthesis	T
	Airborne camera: RGB images	10cm, 21 classes, 100 images/class	[46]	Spatial	I quantification	T
2	Optical: MODIS LST, NDVI	0.05º, 5.5 years, 14-day	[47]	Temporal	I quant., PDF comparison	H, MI
	Passive MW: SMOS SM, VOD	25km, 5.5 years, daily	[48]	Temporal	I quant., PDF comparison	H, MI
3	Obs. & Sim.: E, SH, Precip	0.083º, 10 years, monthly, global	[49]	Spatio-Temporal	I quantification	I, H

imagery. Second, we quantify the information content of several variables describing the soil-vegetation status, and investigate the temporal scales leading to maximum shared information for the detection and precursors of anomalies such as droughts. Finally, we explore the challenging problems of IT measure estimates and the quantification of the spatio-temporal information tradeoff in global Earth products. Table 3 summarizes the experiments in terms of measures, applications and data/simulations used.

3.1 Experiment 1: Gaussianization in remote sensing data

This first set of experiments considers the use of RBIG for standard remote sensing image processing. We will show the performance of RBIG in hyperspectral, very high resolution and radar imagery, and for several applications: joint (multivariate) Gaussianization, data synthesis and information estimation.

3.1.1 Gaussianization of radar images

The first part of the experiment focuses on analysis of radar imagery. Data used here was collected in the Urban Expansion Monitoring (UrbEx) ESA-ESRIN DUP project [44]. Results from UrbEx project were used to perform the analysis of the selected test sites and for validation purposes. We consider an ERS-2 SAR pair selected with perpendicular baselines between 20 and 150 m in order to obtain the interferometric coherence from each complex SAR image pair. The corresponding pair (I_1, I_2) of the SAR backscattering intensities (0-35 days) were stacked for analysis, Figure 4[left]. The relation between the intensity features is strongly nonlinear and non-Gaussian and shows a large dispersion, see Figure 4[a]. The total correlation, T is computed with RBIG for the original domain is $T = 0.0929$ bits. A standard approach in SAR image (pre)processing consists of noise removal and marginal Gaussianization, which

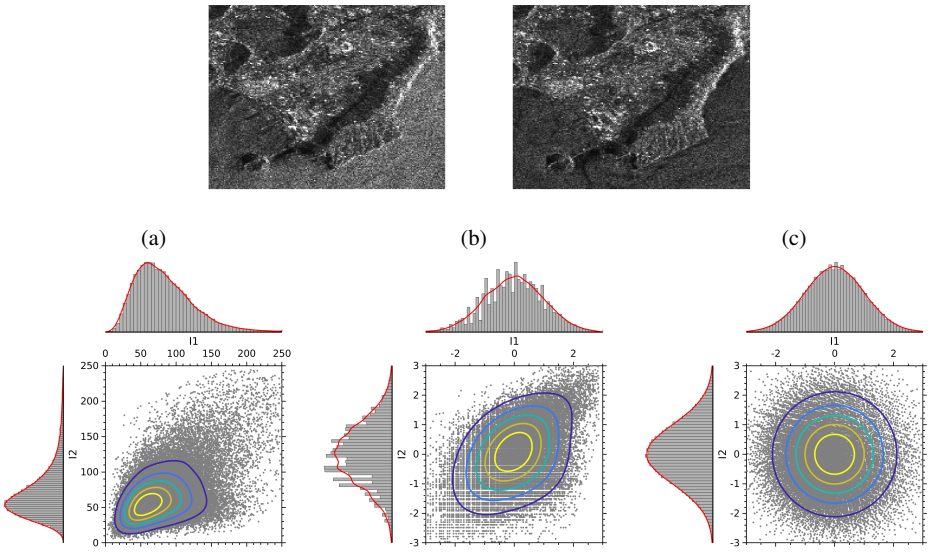


Figure 4: Radar image processing. We illustrate the Gaussinization of 2D radar data comprised of a pair (I1, I2) of ERS-2 SAR backscattering intensities. The joint distribution is non-Gaussian (a) and preprocessing before applying any algorithm is generally convenient. The standard marginal Gaussiaization (b) does not achieve a full spherical (joint) Gaussian, unlike the RBIG transformation (c).

can address these problems only partially. This marginal Gaussianization cannot deal with the saturation for high and low signal values, Figure 4[b]. A multivariate Gaussianization leads to a fully Gaussian density, Figure 4[c]. This is confirmed by the estimated total correlation $T = 0.0095$ bits as it is less than the marginally Gaussianized data.

3.1.2 Synthesizing hyperspectral images

To show the capabilities of the method to deal with high dimensional data here we consider hyperspectral image processing. We took the standard AVIRIS Indian Pines data set [45], where the data has spectrally redundancy and complex joint distrubtions. The image contains 200 spectral channels, which are considered here as the (very high) input dimensionality. We learned a Gaussianization transform leading to a multivariate Gaussian domain of 200 dimensions spectral bands. Then we sampled from a multivariate Gaussian $n = 10^6$ samples of 200 dimensions, and inverted them back to the spectral domain. RBIG can be used this way to generate synthetic spectra easily. Figure 5 (a) shows the original and the synthesized spectra. This shows how the proposed method allows us to generate/synthesize seemingly spectral distributions even in such a high-dimensional setting.

Multispectral, radar and hyperspectral data exhibit complex nonlinear relations in high dimensional spaces. RBIG maps the data to a convenient Gaussian domain that allows to synthesize new data and quantify information

Figure 5 (a) shows the original and the synthesized spectra. This shows how the proposed method allows us to generate/synthesize seemingly spectral distributions even in such a high-dimensional setting. In addition, figure 5 (b) and (c) show corner plots illustrating the joint distributions between various spectral bands (10, 20, 50, 100, and 150). We see that the marginal and joint distributions for the generated spectra by RBIG in (c) are very similar to the real data in (b) across all pairwise band combinations. This is important to highlight that some of the most widely used methods such as PCA would be able to replicate figure 5(a) with a good approximate mean and standard deviation but they would not be able to replicate figure 5(d) where *all joint distributions* are approximately Gaussian.

3.1.3 Information and redundancy in high resolution images

Very high resolution images are constantly acquired with the new generation of sensors, both on airborne and spaceborne platforms. A systematic analysis of the images is necessary. Machine learning, and deep learning in particular, has

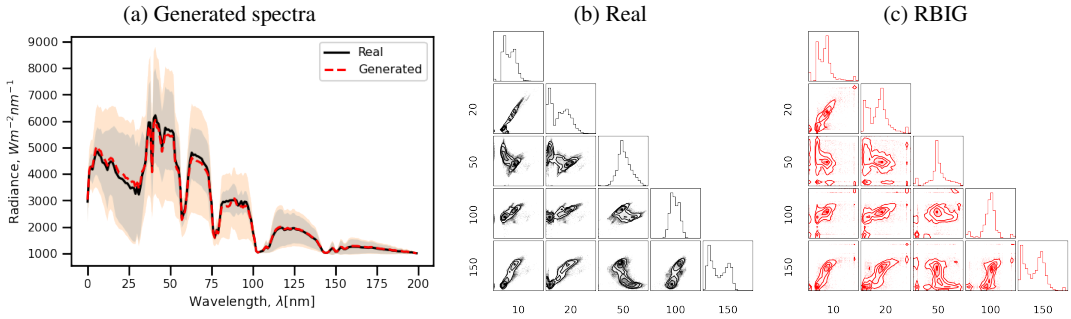


Figure 5: Gaussianization and synthesis of hyperspectral data using RBIG. In (a) we show the mean and standard deviation spectrum for the 21000 real pixels (mean = black, standard deviation = darker shade) and the 1 million pixels generated synthetically (mean = red, standard deviation = lighter shade) using RBIG. In (b) and (c) we show the marginal and joint distributions of 10, 20, 50, 100, 150 spectral bands for the real data and data generated with RBIG respectively.

led to an important leap in classification accuracy. However, owing to the wealth of data and the diversity, it becomes necessary to design algorithms that exploit most of the information content of images, both in terms of relevant features and examples.

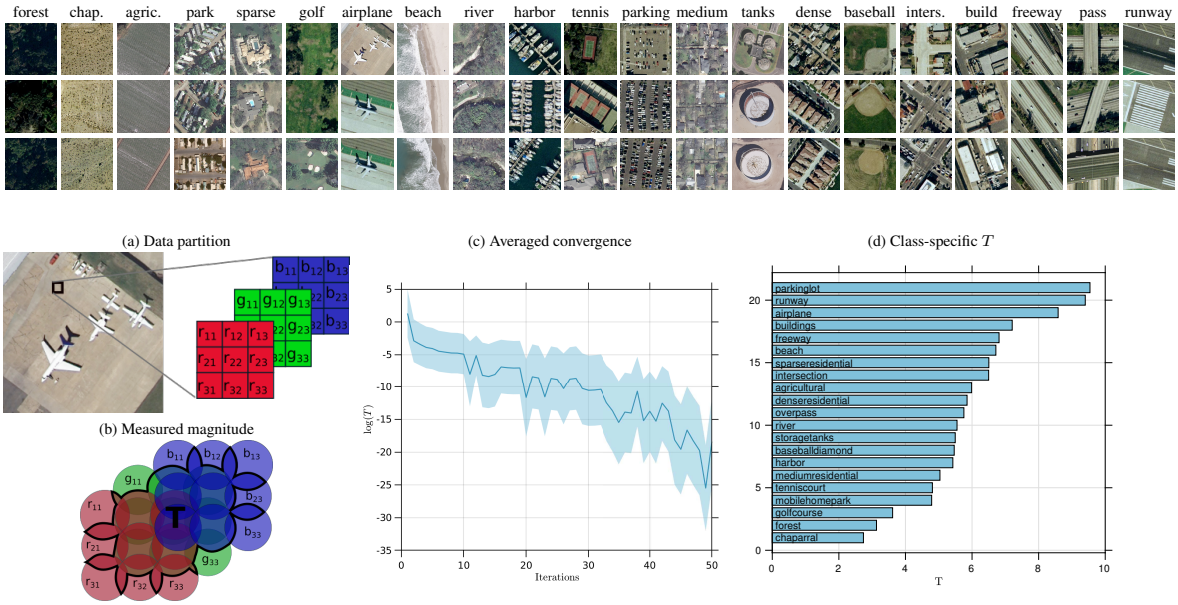


Figure 6: Estimation of total correlation, T in very high resolution aerial imagery. Top: Three illustrative images for each of the 21 classes in the database, ranked according to their estimated T. Bottom: (a) Each image is decomposed in 3×3 patches with three channels (rgb), making samples of 27 dimensions. (b) We measure how much overlapping there is between the information content (i.e. the total correlation) of these 27 dimensions for each class. We show a Venn diagram to illustrate the measured information following the same criteria as in Fig.3. (c) Average total correlation computed iteratively for the different 21 class-specific RBIG models over 50 iterations, with the mean T (solid) and the T standard deviation (shaded) over all models. Convergence is achieved very rapidly for all classes (note the log-scale). (d) ranked T per class computed from the RBIG models.

Here we validate RBIG to estimate total correlation (multi-information) in a set of aerial scenes collected in the UCMerced data set [46]. The data set contains manually extracted images from the USGS National Map Urban Area Imagery collection from 21 aerial scene categories, with 1-ft/pixel resolution. The data set contains highly overlapping classes and has 100 images per class, see some examples per class in Fig. 6[top]. We extracted color patches of size $3 \times 3 \times 3$ from each image, which yielded a total of 6499950 27-dimensional feature vectors per class. Then, we developed a Gaussianization transformation for each class individually and computed the (spatio-spectral) T using RBIG, see Fig. 6[bottom left]. We show in Fig. 6[bottom right] the average and standard deviation of the T evolution through 50 iterations for the 21 classes (note the log-scale) and the total correlation per class. More textured classes like runaways, freeway, buildings and intersections lead to higher T, while rather homogeneous/flat classes like chaparral, agricultural or forests reveal low information content.

3.2 Experiment 2: Information Quantification of Terrestrial Biosphere Variables in Time

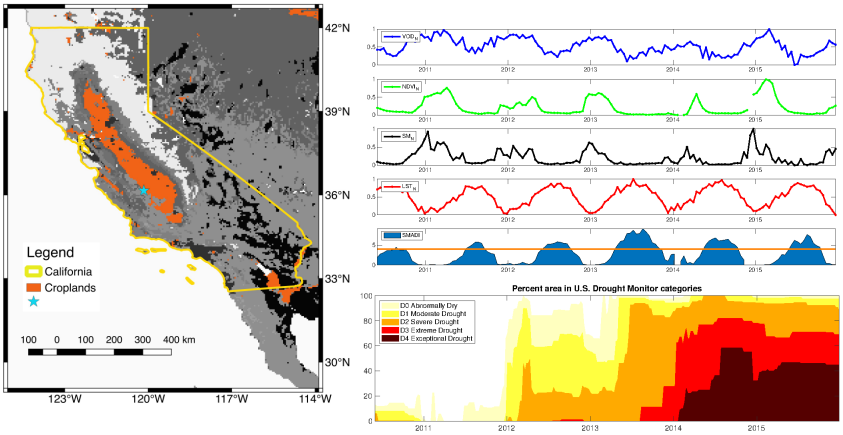


Figure 7: Left: distribution of croplands within California, according to the MODIS IGBP land cover classification. Top right: time series of normalized VOD, NDVI, SM, and LST, as well as the SMADI index [50] obtained at the selected pixel (blue star). SMADI extreme drought category is marked with an orange horizontal line. Bottom right: percent area of California in U.S. drought monitor [51] categories .

According to climate projections, extreme events are likely to intensify and become more frequent over the coming years [52]. The effects of extreme events (such as droughts) are prevalent, not only in the biosphere and atmosphere, but in the anthroposphere too. Drought is a major cause of limited agricultural productivity which accounts for a large proportion of crop losses and annual yield variations throughout the world [53]. Droughts are also being currently placed as direct contributors to social conflicts, migration, and political unrest (e.g. [54]).

There are many studies showing the value of incorporating Earth observation (EO) data for global agricultural systems and applications [55, 56]. Variables such as land surface temperature (LST) and the normalized difference vegetation index (NDVI) derived from optical satellites and, more recently, soil moisture (SM) and vegetation optical depth (VOD) derived from passive microwave sensors are just a few of the many features that can potentially be key for the early detection of drought events [47, 57, 48]. The Soil Moisture Agricultural Drought Index (SMADI) was proposed in [58] to integrate SM with LST and NDVI, showing good agreement with other drought indices and with documented events of drought world-wide [47].

In this experiment, we quantify the information in and between LST, NDVI, SM and VOD variables for the study area of California (only agricultural fields), see figure 7. LST and NDVI are descriptors of the surface temperature and vegetation chlorophyll content, whereas SM and VOD characterize the water content in soils and vegetation [58, 48]. We will also use information measures as a means to evaluate whether it would be worthwhile to include VOD as an additional variable in the SMADI ensemble to characterize droughts. Prior to the analyses, variables were resampled into a common 0.05° grid and biweekly temporal resolution. Details on the data sets are provided in Table 3. Measures are conducted for years 2010-2011 and years 2014-2016 separately, which are representative of non-drought and drought conditions in the study region (see Figure 7).

We focus here in computing multivariate information theory measures in a temporal feature setting, in which previous time steps are included as input features. So for example, 1 input feature includes the current time stamp, 2 input features includes the current time stamp and the time stamp 14 days previously, and so on. This allows us to investigate the temporal scales that maximize the shared information among the remotely-sensed variables. This is particularly relevant for droughts, since there is a time lag between soil/climatic conditions (e.g. represented here by SM, LST) and the plant response (e.g. described by NDVI and VOD), which varies in the literature from two or three weeks up to three months [59].

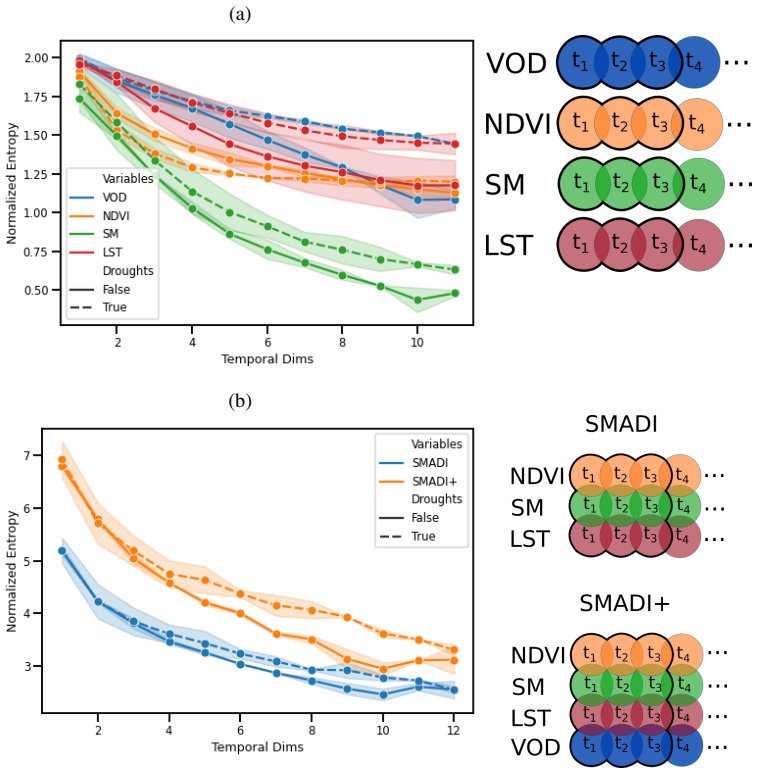


Figure 8: (a) Compares the Entropy for VOD, LST, NDVI and SM individually against the number of temporal dimensions considered. (b) Compares the contribution in entropy of VOD on the joint multidimensional variables integrated in SMADI [LST, NDVI, SM] and SMADI+ [LST, NDVI, SM, VOD] and how it changes as we include more temporal dimensions. The solid lines are mean estimates and the shaded regions are the variance estimates for the non-drought years (2010-2011) and the drought years (2014-2015). Next to each graphic we show a Venn diagram to illustrate the measured information for 3 temporal dimensions as an example, following the same criteria as in Fig.3.

How much information is adding a particular variable for drought monitoring? What temporal scales are the most informative? RBIG can answer these questions explicitly in bits

The amount of expected information H for each of the four variables and how it changes as we include more temporal dimensions is analyzed in Figure 8 (a). Entropy will always increase with more features. So the entropy shown here has been normalized by the total amount of features present which allows us to quantify the amount of *entropy per feature*. It can be seen that the amount of entropy for VOD is the highest in all temporal settings, closely followed by LST. All variables decrease in entropy as we add more temporal features. NDVI saturates at around 1.5 bits whereas the other variables have a steady smooth decline. We can also see that LST and VOD show the largest difference between drought and non-drought years, and that the difference is largest as we increase the temporal dimension. This result suggests that LST and VOD observed during longer periods could be more useful in detecting droughts. Figure 8 (b) shows that VOD increases the amount of

expected information when added to the SMADI variable ensemble in all the temporal settings considered, suggesting that it would be worthwhile to include VOD in agricultural drought studies. Results indicate operational settings of vegetation monitoring could benefit from synergistic approaches that allow including multi-sensor multi-dimensional variables, in particular under stress and disturbances such as agricultural droughts.

The MI of every pair of multidimensional variables was analyzed to investigate the relation and redundancy between them as well as the optimal time scales to combine them. Note standard measures for pairwise comparison such as Pearson's correlation are restricted to one temporal dimension and hence do not allow exploring these scales. The MI scores obtained for LST relations are shown in Figure 9. Interestingly, it shows that LST-NDVI and LST-VOD show an increase in mutual information up to about 2-4 temporal dimensions and then it saturates. This result suggests that a period of about 1-2 months is needed to capture the soil-plant status with the remotely-sensed variables analyzed in our study region. The curves are relatively similar irregardless of whether it is a drought year or not, and the spread of values for the drought years is considerably reduced for all variables and especially for VOD. This could be related to a reduced variability (limited range of values) under drought episodes, but further studies are needed to confirm this. We also observed that MI is consistently low between SM and all variables with any number of temporal dimensions, and is also low between NDVI and VOD, highlighting the value of combining optical and microwave variables for vegetation/land monitoring.

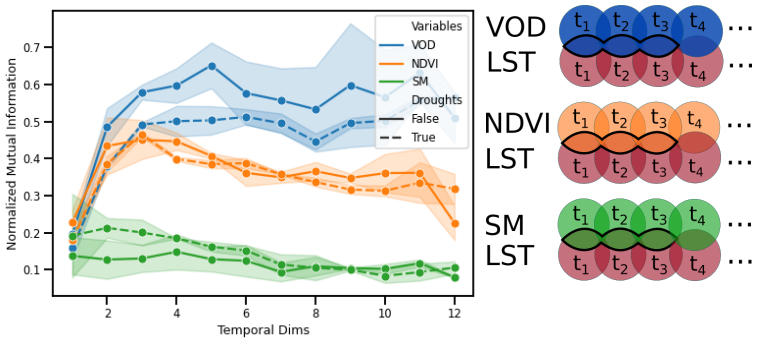


Figure 9: Mutual information between pairs of multidimensional variables: LST-VOD, LST-NDVI, and LST-SM. The solid lines are mean estimates and the shaded regions are the variance estimates for the non-drought years (2010-2011) and the drought years (2014-2015). The Venn diagram illustrates the measured information for 3 temporal dimensions as an example, following the same criteria as in Fig.3.

3.3 Experiment 3: Information in Spatial-Temporal Earth data

3.3.1 Data

For our experiments we used observational and model simulated variables from the Earth Science Data Lab (ESDL) [49][5], which is a platform that provides an opportunity for data-centric processing methodologies. The analysis-ready data-cube contains and harmonizes more than 40 variables relevant to monitor key processes of the terrestrial land-surface and atmosphere. Data exhibit clear spatial-temporal relations, which need to be taken into account to properly convey and quantify information. Figure 10 illustrates how we represent this spatial-temporal relations as inputs given a single variable. Here we focus on three key land-surface variables: precipitation, sensible heat and evaporation, which are outlined below:

- *Precipitation, $Precip$.* This is a fundamental variable in land-atmosphere processes. The collected data comprises the period 1980–2015, and comes from the Global Precipitation Climatology Project (GPCP) [60, 61].
- *Sensible heat, SH.* These data comprises 2001–2012, and was generated by training an ensemble of machine learning algorithms with eddy covariance data from FLUXNET and satellite observations in a cross-validation approach, regressions from these observations to different kinds of carbon and energy fluxes were established and used to generate data sets with a spatial resolution of 5 arc-minutes and a temporal resolution of 8 days. The H resembles the sensible heat flux from the surface and is expressed in [W m^{-2}] [62].

[5] https://www.earthsystemdatalab.net/

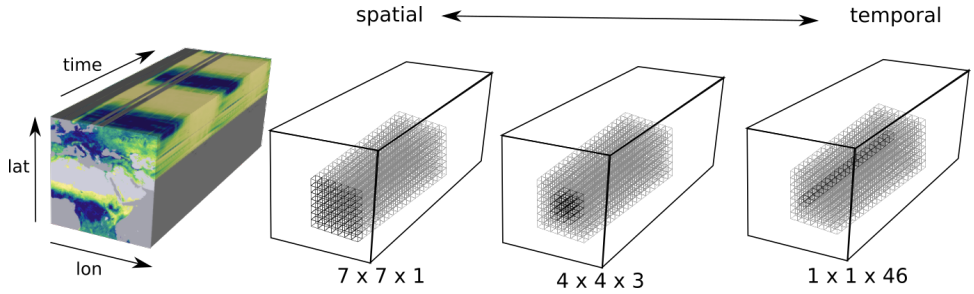


Figure 10: This figure illustrates an example decomposition of the Earth science data cube (ESDC)[49] into different spatial-temporal configurations ranging from completely spatial to completely temporal. The $7 \times 7 \times 1$ spatial configuration is all spatial pixels and no temporal pixels; this is very similar to spatial patches. The $1 \times 1 \times 46$ configuration is all temporal pixels but no spatial pixels which is essentially a time series. The $4 \times 4 \times 3$ configuration is a mixture of spatial and temporal pixels. Through-out this article we see different notions of spatial-temporal representation of the ESDC data.

- *Evaporation, E.* These data covers 2001–2011, and builds on the Global Land Evaporation Amsterdam Model (GLEAM), which consists on a set of algorithms that separately estimate the different components of land evaporation using input forcing data sets from reanalyses, optical and microwave satellites and other merged sources. The model itself consists of four modules: potential evaporation (Priestley and Taylor equation), interception (Gash analytical model), soil (multilayer soil model plus data assimilation) and stress (semi-empirical). The data are sampled on a grid of 0.25° and have a daily temporal coverage [63, 64].

The data is organized in a 4-dimensional data cube $\mathbf{x}(u, v, t, k)$ involving (latitude, longitude) spatial coordinates (u, v), time sampling t, and the variable k. The available data is provided at two spatial resolutions (0.083^o and 0.25^o) and at a temporal resolution of 8 days, spanning the years 2001-2011. In our experiments, we focus on the lower resolution products and on the period 2008-2010.

3.3.2 Spatial-Temporal Analysis

The considered variables (precipitation, sensible heat, evaporation) are fully coupled. Moisture and precipitation interactions are vastly modulated by both land-atmosphere exchanges and large-scale atmospheric circulation. Nevertheless, before understanding variable relations, it is important to *identify when and where* individual variables are expressive. This may help in assessing the coupling mechanisms between variables and improve Earth system models.

What are the optimal spatial and temporal scales to exploit each variable in the coupled land-atmosphere system? RBIG sheds light on uncertainty quantification and information in multidimensional spatio-temporal Earth data cubes

The question we want to address in this experiment is what are the optimal (in information terms) spatial and temporal scales to exploit each variable's information. Using RBIG, we show here that the ratio of spatial-temporal neighbouring pixels giving the most amount of information can be explicitly calculated. We used RBIG to calculate the entropy H for the aforementioned variables under different spatial-temporal configurations (fully temporal, spatio-temporal and fully spatial) as well as the corresponding information $I(\mathbf{x})$ for each time pixel and variable.

Figure 11 shows the entropy for the different variables and configurations following the same procedure as in [65] (and used in experiment §3.1.3 too in the spatial domain only). Essentially we formed cubes with the same dimensionality but different spatio-temporal configuration and computed the entropy values for each of them. We chose several configurations ranging from a ratio of purely spatial (ratio=0) up to purely temporal (ratio=1). We also looked at different configurations for the amount of spatial-temporal dimensions used, e.g. a maximum of 4 dimensions up to a maximum of 49 dimensions (temporally, this is approximately 1 year). Notice how each variable has a different spatial-temporal relationship with entropy, but in general temporal configurations (ratio=1) convey more information than purely spatial (ratio=0) for all the considered variables. Trends are clear in particular for precipitation, where incorporating temporal information for any amount of dimensions has higher expected information. For sensible heat and evaporation the entropy paths are similar, and reveal a fast increase in entropy for particular spatio-temporal configurations (ratio$\sim$0.8). These results suggest different *optimal* (in information terms) time and space scales for different variables, which may have implications in further analyses and applications.

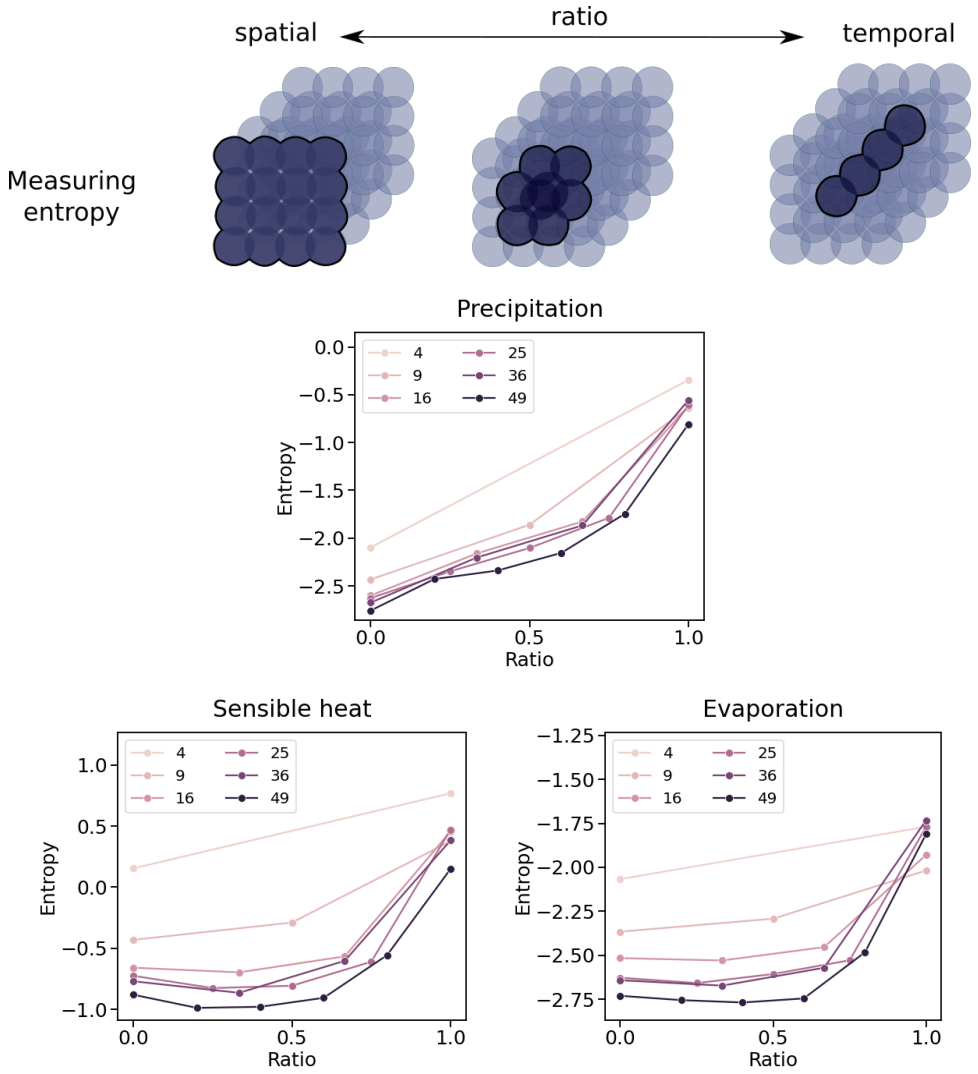


Figure 11: Measuring entropy for different spatio-temporal configurations in the ESDC variables. The **top row** information theory Venn diagram representation of Fig. 10 and how this relates to measuring entropy: the expected uncertainty. In the **middle and bottom rows**, we show how the measured entropy for precipitation, sensible heat and evaporation from the Earth science data cube [49] changes with different spatial-temporal representations, ranging from fully spatial (ratio = 0) and fully temporal representation (ratio = 1).

Using the same data configurations we have computed the information content of each sample following the procedure described in section 2.3.1. This help us to visualize the regions with more and less information. We show in Fig. 12 the results of a spatio-temporal analysis of the information content of all three variables. In regions where we expect pronounced seasonal patterns, the information (complexity) is apparently high in fully temporal configurations as the seasonal cycle controls ecosystem dynamics. Actually, seasonal (temporal) modes are of lower informative content in the spatial domain, as they are mainly driven by solar forcing. The information values tend to be higher in tropical regions, whereas arid regions show low-complexity (low-information) patterns. Let us now look in deeper detail at the different spatio-temporal configurations and their information patterns.

Global patterns in rainfall are traditionally related to a strong seasonality, dominated by the position of the Inter-Tropical Convergence Zone (ITCZ) in the tropics, and the El Niño-La Niña cycles, which occur irregularly at intervals of 2-7 years. Spatial information generally dominates with high probability in the Amazonia and the tropics and with low information in desertic areas (e.g. California, Arabian peninsula and central Australia). As we quantify information in spatio-temporal configurations, more clear patterns of low information (e.g. Australia) and high probability (e.g. east-west US gradient) emerge [66]. Studying precipitation in the fully temporal configuration translates into a clear ruling of the winter season in Amazonia, Indonesia, as well as northern Europe. Yet, a comparison of temporal vs. spatial information in Fig. 12[bottom row] reveals that spatial information dominates in desertic areas (e.g. Australia, Iberian peninsula, Sahara, Mexico) which are reasonably independent of time, and temporal information dominates in Sahel (Savanna), northern latitudes and SW China, which are generally characterized by high rain factors, seasons and moisture.

Transfer of sensible heat SH into the air is dependent on the temperature gradient between the surface and the air above. Patterns of the information of sensible heat SH stand out clearly. While the (fully) spatial information dominates in the Northern hemisphere, the (fully) temporal information patterns appear in the tropics where rainfall is present over larger regions and seasons. The global spatial distribution of SH information shows the largest values in subtropical dry regions where available energy is preferentially partitioned to sensible heat rather than latent heat [67], and seem to be anti-correlated with the amplitude of the mean seasonal cycle. These results reveal a maximum information of SH in the tropical and subtropical deserts, where the high surface temperature conducts much heat into the air above, and the lowest near the poles where the surface temperatures are much lower. Information is mainly concentrated in the tropics too, and show similar patterns to precipitation with the exception of clear spatial information in the Indian peninsula. Evaporation maps of information reveal that the spatial information dominates in deserts and dry regions where evaporation is limited, while temporal information (more interannual variability) resides in Northern latitudes. This is mainly due to the low temperatures and radiation which relates to little evaporation all year round. The temperate areas show increased evaporation information in both purely spatial and temporal configurations, coinciding with increasing temperatures over ground moistened by winter rains. Cooler winter temperatures in Southern hemisphere reduce evaporation, which is also captured in the spatial-vs-temporal divergent maps. Note that in very dry regions information is higher (lower evaporative fraction), conversely for very humid regions, in agreement with [67].

4 Conclusions

This paper introduces a Gaussianization method and illustrates how to use it for multivariate density estimation in the context of Earth system science. The problem is highly relevant with the advent of all kinds of Earth data, both remotely sensed and in situ observations, novel products and model simulations. Density estimation is a long-standing unresolved problem in statistics and machine learning, mainly because of the curse of dimensionality. Besides, the data in remote sensing and geosciences pose additional challenging problems for PDF estimation: high dimensional data, nonlinear feature relations, many noise sources, and distinct spatial-temporal structures.

The Gaussianization method allows to simplify the problem by learning an invertible transformation of the data distribution to a multivariate Gaussian domain where features are independent. This not only makes the PDF estimation well-posed, but also allows us to estimate key information theoretic measures on multivariate datasets: information, entropy, total correlation and mutual information. We showed that the methodology can deal with high dimensionality and a high volume of data, and is simple to use and apply in spatio-temporal domains. We provide source code for the interested reader.

We showed empirical evidence of performance in several Earth system data analysis problems, using a wide diversity of data (multispectral, hyperspectral, SAR, as well as global products from both satellites and Earth system models), and addressed the key problems of information estimation, redundancy, and synthesis. Results confirmed the validity of the method, for which we anticipate a wide use and adoption.

The framework enables us to tackle all applications involving a PDF estimation; from data classification to denoising and coding, which were not treated in this paper. The methodology also allows to compute other interesting IT measures, such as Kullback-Leibler divergence and conditional independence, which will be a subject of future research.

5 Acknowledgements

This research was funded by the European Research Council (ERC) under the ERC-Consolidator Grant 2014 Statistical Learning for Earth Observation Data Analysis. project (grant agreement 647423). J.E.J. thanks the European Space Agency (ESA) for support via the Early Adopter Call of the Earth System Data Lab project; M.D.M. thanks the ESA

for the long-term support of this initiative. Additional support was provided by the Project RTI2018-096765-A-100 (MCIU/AEI/FEDER, UE).

References

[1] W. Buermann, J. Dong, X. Zeng, R. B. Myneni, and R. E. Dickinson, "Evaluation of the utility of satellite-based vegetation leaf area index data for climate simulations," *Journal of Climate*, vol. 14, no. 17, pp. 3536–3550, 2001.

[2] R. H. Moss, J. Edmonds, K. A. Hibbard, M. Manning, S. Rose, D. van Vuuren, T. Carter, S. Emori, M. Kainuma, T. Kram, G. Meehl, J. Mitchell, N. Nakicenovic, K. Riahi, S. Smith, R. Stouffer, A. Thomson, J. Weyant, and T. Wilbanks, "The next generation of scenarios for climate change research and assessment," *Nature*, vol. 463, pp. 747–756, 2010.

[3] J. T. Overpeck, G. A. Meehl, S. Bony, and D. R. Easterling, "Climate data challenges in the 21st century," *Science*, vol. 331, no. 700, 2011.

[4] V. Eyring, P. M. Cox, G. M. Flato, P. J. Gleckler, G. Abramowitz, P. Caldwell, W. D. Collins, B. Gier, A. D. Hall, F. M. Hoffman, G. C. Hurtt, A. Jahn, C. D. Jones, S. A. Klein, J. Krasting, L. Kwiatkowski, R. Lorenz, E. Maloney, G. A. Meehl, A. Pendergrass, R. Pincus, A. C. Ruane, J. L. Russell, B. Sanderson, B. Santer, S. C. Sherwood, I. Simpson, R. Stouffer, and M. Williamson, "Taking climate model evaluation to the next level," *Nature Climate Change*, 2018.

[5] M. Reichstein, G. Camps-Valls, B. Stevens, J. Denzler, N. Carvalhais, M. Jung, and Prabhat, "Deep learning and process understanding for data-driven Earth System Science," *Nature*, vol. 566, pp. 195–204, Feb 2019.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory, 2nd Edition*. Wiley, 2006.

[7] G. A. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Trans. Inf. Theor.*, vol. 45, no. 4, pp. 1315–1321, Sep. 2006.

[8] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, p. 066138, Jun 2004.

[9] Q. Wang, S. Kulkarni, and S. Verdú, "A nearest-neighbor approach to estimating divergence between continuous random vectors," ser. IEEE International Symposium on Information Theory - Proceedings, 12 2006, pp. 242–246.

[10] N. Leonenko, L. Pronzato, and V. Savani, "A class of rényi information estimators for multidimensional densities," *Annual Statistics*, vol. 36, no. 5, pp. 2153–2182, 10 2008.

[11] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[12] F. Pérez-Cruz, "Estimation of information theoretic measures for continuous random variables," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1257–1264.

[13] S. Paul and D. N. Kumar, "Spectral-spatial classification of hyperspectral data with mutual information based segmented stacked autoencoder approach," *ISPRS journal of photogrammetry and remote sensing*, vol. 138, pp. 265–280, 2018.

[14] A. G. Konings, K. A. McColl, M. Piles, and D. Entekhabi, "How many parameters can be maximally estimated from a set of measurements?" *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 5, pp. 1081–1085, 2015.

[15] A. Marinoni and P. Gamba, "Unsupervised data driven feature extraction by means of mutual information maximization," *IEEE Transactions on Computational Imaging*, vol. 3, no. 2, pp. 243–253, 2017.

[16] J. Zhang, M. Zareapoor, X. He, D. Shen, D. Feng, and J. Yang, "Mutual information based multi-modal remote sensing image registration using adaptive feature weight," *Remote Sensing Letters*, vol. 9, no. 7, pp. 646–655, 2018.

[17] S. Prasad and L. M. Bruce, "Hyperspectral feature space partitioning via mutual information for data fusion," in *2007 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2007, pp. 4846–4849.

[18] L.-Y. Zhao, B.-Y. Lü, X.-R. Li, and S.-H. Chen, "Multi-source remote sensing image registration based on scale-invariant feature transform and optimization of regional mutual information," *Acta Physica Sinica*, vol. 64, no. 12, p. 124204, 2015.

[19] S. Chen, X. Li, L. Zhao, and H. Yang, "Medium-low resolution multisource remote sensing image registration based on sift and robust regional mutual information," *International Journal of Remote Sensing*, vol. 39, no. 10, pp. 3215–3242, 2018.

[20] X. Xu, X. Li, X. Liu, H. Shen, and Q. Shi, "Multimodal registration of remotely sensed images based on jeffrey's divergence," *ISPRS journal of photogrammetry and remote sensing*, vol. 122, pp. 97–115, 2016.

[21] B. L. Ruddell, N. A. Brunsell, and P. Stoy, "Applying information theory in the geosciences to quantify process uncertainty, feedback, scale," *Eos, Transactions American Geophysical Union*, vol. 94, no. 5, pp. 56–56, 2013.

[22] G. Papamakarios, "Neural density estimation and likelihood-free inference," *PhD Thesis*, vol. abs/1910.13233, 2019.

[23] L. Ardizzone, J. Kruse, C. Rother, and U. Köthe, "Analyzing inverse problems with invertible neural networks," 2019.

[24] D. J. Rezende, G. Papamakarios, S. Racanière, M. S. Albergo, G. Kanwar, P. E. Shanahan, and K. Cranmer, "Normalizing flows on tori and spheres," *ArXiv*, vol. abs/2002.02428, 2020.

[25] S. S. Chen and R. A. Gopinath, "Gaussianization," in *NIPS*, 2000.

[26] V. Laparra, G. Camps-Valls, and J. Malo, "Iterative gaussianization: From ica to random rotations," *IEEE Transactions on Neural Networks*, vol. 22, pp. 537–549, 2011.

[27] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," *ICLR*, vol. abs/1511.06281, 2016.

[28] C. Meng, Y. Song, J. Song, and S. Ermon, "Gaussianization flows," *ArXiv*, vol. abs/2003.01941, 2020.

[29] E. T. Nalisnick, A. Matsukawa, Y. W. Teh, D. Görür, and B. Lakshminarayanan, "Hybrid models with deep and invertible features," *ArXiv*, vol. abs/1902.02767, 2019.

[30] C. M. Bishop and N. M. Nasrabadi, "Pattern recognition and machine learning," *J. Electronic Imaging*, vol. 16, p. 049901, 2007.

[31] G. M. Goerg, "The lambert way to gaussianize heavy-tailed data with the inverse of tukey's h transformation as a special case," 2015.

[32] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 7511–7522.

[33] G. E. P. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 26, no. 2, pp. 211–252, 1964.

[34] J. H. Friedman, "Exploratory projection pursuit," vol. 82, 1987, pp. 249–266.

[35] D. I. Inouye and P. Ravikumar, "Deep density destructors," in *ICML*, 2018.

[36] P. Jaini, K. A. Selby, and Y. Yu, "Sum-of-squares polynomial flow," ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 3009–3018.

[37] G. Liu, Y. Liu, M. Guo, P. Li, and M. Li, "Variational inference with gaussian mixture model and householder flow," *Neural networks : the official journal of the International Neural Network Society*, vol. 109, pp. 43–55, 2019.

[38] J. M. Tomczak and M. Welling, "Improving variational auto-encoders using householder flow," *ArXiv*, vol. abs/1611.09630, 2016.

[39] J. Cardoso, "Dependence, correlation and gaussianity in independent component analysis," *J. Mach. Learn. Res.*, vol. 4, pp. 1177–1203, 2003.

[40] C. E. Shannon, "The mathematical theory of communication," vol. 27, 1948, pp. 379–423.

[41] M. S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM J. Res. Develop.*, vol. 4, pp. 66–82, 1960.

[42] M. Studený and J. Vejnarová, "The multiinformation function as a tool for measuring stochastic dependence," in *Proc. NATO Adv. Study Inst. Learn. Graph. Models*. Kluwer, 1998, pp. 261–297.

[43] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, p. 066138, Jun 2004.

[44] L. Gómez-Chova, D. Fernández-Prieto, J. Calpe, E. Soria, J. Vila-Francés, and G. Camps-Valls, "Urban monitoring using multitemporal sar and multispectral data," *Pattern Recognition Letters, Special Issue on "Pattern Recognition in Remote Sensing"*, vol. 27, no. 4, pp. 234–243, 2006.

[45] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3," Sep 2015.

[46] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 270–279.

[47] N. Sánchez, Á. González-Zamora, J. Martínez-Fernández, M. Piles, and M. Pablos, "Integrated remote sensing approach to global agricultural drought monitoring," *Agric. For. Meteorol.*, 2018.

[48] R. Fernandez-Moran, A. Al-Yaari, A. Mialon, A. Mahmoodi, A. Al Bitar, G. De Lannoy, N. Rodriguez-Fernandez, E. Lopez-Baeza, Y. Kerr, and J.-P. Wigneron, "SMOS-IC: An Alternative SMOS Soil Moisture and Vegetation Optical Depth Product," *Remote Sensing*, vol. 9, no. 5, p. 457, May 2017.

[49] M. Mahecha, F. Gans, G. Brandt, R. Christiansen, S. Cornell, N. Fomferra, G. Kraemer, J. Peters, G. Camps-Valls, J. Donges, W. Dorigo, L. Estupinan-Suarez, V. Gutierrez-Velez, M. Gutwin, M. Jung, M. Londono, D. Miralles, P. Papastefanou, and M. Reichstein, "Earth system data cubes unravel global multivariate dynamics," *Earth System Dynamics*, vol. 11, pp. 201–234, Feb 2020.

[50] Ángel González-Zamora, N. Sánchez, and M. Piles, "Global Soil Moisture Agricultural Drought Index (SMADI)," Jun. 2019, This study was supported by: -The Spanish Ministry of Economy and Competitiveness, MINECO (Project AYA2012-39356-C05). -The Spanish Ministry of Science, Innovation and Universities (Projects ESP2015-67549-C3-3 and ESP2017-89463-C3-3-R) -The Castilla y León Regional Government (Project SA007U16). -The European Regional Development Fund (ERDF). [Online]. Available: https://doi.org/10.5281/zenodo.3247649

[51] "U.S. Drought Monitor," national Drought Mitigation Center, U.S. Department of Agriculture, and National Oceanic and Atmospheric Association. [Online]. Available: https://droughtmonitor.unl.edu/

[52] J. Zscheischler, M. D. Mahecha, S. Harmeling, and M. Reichstein, "Detection and attribution of large spatiotemporal extreme events in Earth observation data," *Ecological Informatics*, vol. 15, pp. 66–73, 2013.

[53] J. S. Boyer, "Plant Productivity and Environment," *Science (80-.).*, vol. 218, no. 4571, pp. 443–448, 1982.

[54] C. P. Kelley, S. Mohtadi, M. A. Cane, R. Seager, and Y. Kushnir, "Climate change in the Fertile Crescent and implications of the recent Syrian drought," *Proc. Natl. Acad. Sci.*, vol. 112, no. 11, pp. 3241–3246, 2015.

[55] S. Fritz, L. See, J. C. L. Bayas, F. Waldner, D. Jacques, I. Becker-Reshef, A. Whitcraft, B. Baruth, R. Bonifacio, J. Crutchfield, F. Rembold, O. Rojas, A. Schucknecht, M. V. der Velde], J. Verdin, B. Wu, N. Yan, L. You, S. Gilliams, S. Mücher, R. Tetrault, I. Moorthy, and I. McCallum, "A comparison of global agricultural monitoring systems and current gaps," *Agric. Syst.*, vol. 168, pp. 258–272, 2019.

[56] M. Weiss, F. Jacob, and G. Duveiller, "Remote sensing for agricultural applications: A meta-review," *Remote Sens. Environ.*, vol. 236, p. 111402, 2020.

[57] S. Sadri, E. F. Wood, and M. Pan, "Developing a drought-monitoring index for the contiguous US using SMAP," *Hydrol. Earth Syst. Sci.*, vol. 22, no. 12, pp. 6611–6626, 2018.

[58] N. Sánchez, Á. González-Zamora, M. Piles, and J. Martínez-Fernández, "A new Soil Moisture Agricultural Drought Index (SMADI) integrating MODIS and SMOS products: A case of study over the Iberian Peninsula," *Remote Sensing*, vol. 8, no. 4, 2016.

[59] G. P. Petropoulos and T. Islam, *Remote Sensing of Hydrometeorological Hazards*. CRC Press, 2017.

[60] R. F. Adler, G. J. Huffman, A. Chang, R. Ferraro, P.-P. Xie, J. Janowiak, B. Rudolf, U. Schneider, S. Curtis, D. Bolvin, A. Gruber, J. Susskind, P. Arkin, and E. Nelkin, "The Version-2 Global Precipitation Climatology Project (GPCP) Monthly Precipitation Analysis (1979-Present)," *Journal of Hydrometeorology*, vol. 4, no. 6, pp. 1147–1167, 2003.

[61] G. J. Huffman, R. F. Adler, D. T. Bolvin, and G. Gu, "Improving the global precipitation record: Gpcp version 2.1," *Geophysical Research Letters*, vol. 36, no. 17, 2009.

[62] G. Tramontana, M. Jung, C. R. Schwalm, K. Ichii, G. Camps-Valls, B. Ráduly, M. Reichstein, M. A. Arain, A. Cescatti, G. Kiely, L. Merbold, P. Serrano-Ortiz, S. Sickert, S. Wolf, and D. Papale, "Predicting carbon dioxide and energy fluxes across global fluxnet sites with regression algorithms," *Biogeosciences*, vol. 13, no. 14, pp. 4291–4313, 2016.

[63] B. Martens, D. G. Miralles, H. Lievens, R. van der Schalie, R. A. M. de Jeu, D. Fernández-Prieto, H. E. Beck, W. A. Dorigo, and N. E. C. Verhoest, "Gleam v3: satellite-based land evaporation and root-zone soil moisture," *Geoscientific Model Development*, vol. 10, no. 5, pp. 1903–1925, 2017.

[64] D. G. Miralles, T. R. H. Holmes, R. A. M. De Jeu, J. H. Gash, A. G. C. A. Meesters, and A. J. Dolman, "Global land-surface evaporation estimated from satellite-based observations," *Hydrology and Earth System Sciences*, vol. 15, no. 2, pp. 453–469, 2011.

[65] V. Laparra and R. Santos-Rodriguez, "Spatial/spectral information trade-off in hyperspectral images," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 07 2015, pp. 1124–1127.

[66] S. Tuttle and G. Salvucci, "Empirical evidence of contrasting soil moisture–precipitation feedbacks across the united states," *Science*, vol. 352, no. 6287, pp. 825–828, 2016.

[67] M. Jung, M. Reichstein, H. A. Margolis, A. Cescatti, A. D. Richardson, M. A. Arain, A. Arneth, C. Bernhofer, D. Bonal, J. Chen *et al.*, "Global patterns of land-atmosphere fluxes of carbon dioxide, latent heat, and sensible heat derived from eddy covariance, satellite, and meteorological observations," *Journal of Geophysical Research: Biogeosciences*, vol. 116, no. G3, 2011.

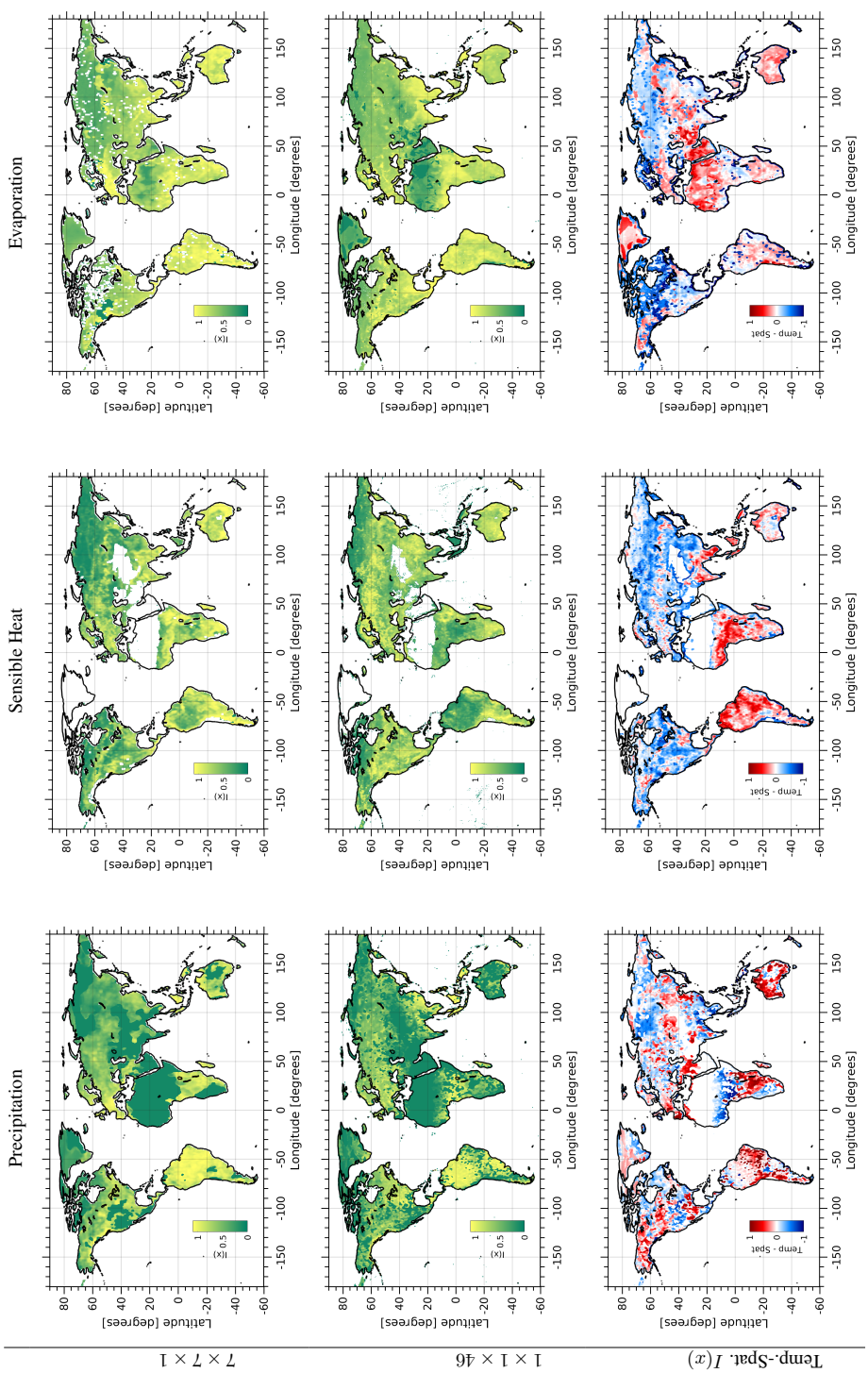


Figure 12: The two first rows show information content maps for precipitation, sensible heat, and evaporation using a fully spatial (7×7 spatial width, 1 temporal length) and a fully temporal (1×1 spatial width, 46 temporal length) configuration. The bottom row shows a divergent map of the trade-off (subtraction) between the information content of fully spatial and fully temporal per each variable.